

Article

An HMM-Based Approach for Cross-Harmonization of Jazz Standards

Maximos Kaliakatsos-Papakostas ^{1,2,*} , Konstantinos Velenis ¹, Leandros Pasiadis ¹, Chrisoula Alexandraki ²  and Emiliós Cambouropoulos ¹

¹ School of Music Studies, Aristotle University of Thessaloniki, 541 24 Thessaloniki, Greece

² Department of Music Technology and Acoustics, Hellenic Mediterranean University, 741 33 Rethymno, Greece

* Correspondence: maximoskp@hmu.gr

Abstract: This paper presents a methodology for generating cross-harmonizations of jazz standards, i.e., for harmonizing the melody of a jazz standard (Song A) with the harmonic context of another (Song B). Specifically, the melody of Song A, along with the chords that start and end its sections (chord constraints), are used as a basis for generating new harmonizations with chords and chord transitions taken from Song B. This task involves potential incompatibilities between the components drawn from the two songs that take part in the cross-harmonization. In order to tackle such incompatibilities, two methods are introduced that are integrated in the Hidden Markov Model and the Viterbi algorithm. First, a rudimentary approach to chord grouping is presented that allows interchangeable utilization of chords belonging to the same group, depending on melody compatibility. Then, a “supporting” harmonic space of chords and probabilities is employed, which is learned from the entire dataset of the available jazz standards; this space provides local solutions when there are insurmountable conflicts between the melody and constraints of Song A and the harmonic context of Song B. Statistical and expert evaluation allow an analysis of the methodology, providing valuable insight about future steps.

Keywords: Hidden Markov Model; melodic harmonization; jazz standards; computational creativity



Citation: Kaliakatsos-Papakostas, M.; Velenis, K.; Pasiadis, L.; Alexandraki, C.; Cambouropoulos, E. An HMM-Based Approach for Cross-Harmonization of Jazz Standards. *Appl. Sci.* **2023**, *13*, 1338. <https://doi.org/10.3390/app13031338>

Academic Editor: Rocco Zaccagnino

Received: 22 December 2022

Revised: 15 January 2023

Accepted: 16 January 2023

Published: 19 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In music, as in other forms of art and creativity, employing transformed versions of ideas and concepts from existing works is common practice. This transformation process often leads to the generation of original works that incorporate novel ideas and concepts. This is emphasized by the infamous aphorism “good artists borrow, great artists steal” that has been attributed to Pablo Picasso. Conceptual Blending Theory (CBT), as initially described by Fauconnier and Turner in [1], describes a process where concepts from two independent spaces are combined creatively to create new spaces that lead to the invention of original concepts in their own right. In music, possible effects of conceptual blending in musical works have been studied extensively by Zbikowski, e.g., in [2–4], while creative computational systems that are based on the mathematical formalization of Goguen [5,6] have been examined, e.g., for conceptual blending of harmonic spaces [7].

Jazz standards have commonly been notated as melodies accompanied by simplified harmonic frameworks of chords (lead sheets, fake books), allowing thus the necessary space for improvisation. Jazz musicians transform a given harmonic reduction not only in terms of voicing, but also by “blending in” ideas from a repertoire of “harmonic clichés” such as chord extensions, chord substitutions, secondary dominants, local modulations, turnarounds, blue chords and so on, drawn from different periods and styles of jazz music. Occasionally jazz performers re-harmonize thoroughly jazz standards, going beyond established lead music settings (e.g., Bill Evans). In this paper, we focus on one specific topic, namely, the creative potential of combining the melody of one jazz standard, Song A

(with its inherent implied harmonic space), with the harmonic space drawn from a second different jazz standard, Song B—creating some form of harmonic single-scope blends [8]. Can the melody-harmony blending of two different jazz standards create original and interesting harmonizations that not only (re)invent some of the known harmonic formulas, but also give rise to novel harmonic realisations?

The specific problem addressed in this paper is the algorithmic generation of new harmonizations of one jazz standard melody based on the harmony of a different jazz standard. A method that allows this type of cross-harmonization is aimed at enabling new possibilities for music exploration emerging from conflicting incompatibilities between two pieces. The inherently improvisational nature of jazz music performance demands developing skills of building tension based on conflicting harmonic constructs. It is thus considered important for teaching and learning music improvisation to provide ideas for the development of such skills. The possibility of harmonic blending presented in this paper may be useful to instructors to design exercises depicting conflicts as new challenges to their students as well as to students during self-practicing. Can such a blending system for melodic harmonization assist jazz performers in their creative practice? What happens in the case where two spaces are incompatible in certain respects?

Two methods have been developed and studied that attempt to resolve impasses that may be caused due to incompatibilities between two jazz standard spaces (e.g., a certain melodic transition in Song A may not be harmonizable by the chord transitions appearing in Song B). The first method generalizes the concept of chords by introducing chord groups, the members of which can be used interchangeably depending on the underlying melody. The second method incorporates an extension of the Viterbi algorithm in Hidden Markov Models [9,10], where incompatibilities between the harmony in Song B and the melody in Song A are resolved by introducing a “support” harmonic space that represents the “background knowledge” of the entire available dataset of jazz standards. This support space enables chord transitions that are not present in Song B, but are compatible with the overall jazz idiom.

Jazz is known for its complex melodies and harmonies, and its improvisational nature. Harmonization is a fundamental aspect of jazz, where multiple musical voices combine to create a rich and complex sound. In recent years, there has been growing interest in using artificial intelligence (AI) to assist the harmonization of melodies in jazz [11,12]. However, one major challenge in this area is the scarcity of properly annotated datasets [13–15], in the case of jazz standards, mainly due to copyright issues with the melodies. Additionally, lead sheets, which are commonly used in jazz to notate melodies, are often not properly annotated with tonality and other structural information, e.g., sections. As a result, AI models have difficulty learning to harmonize melodies in a way that accurately captures the essence of the idiom [16]. This highlights the importance of properly annotated datasets in the development of AI models for musical tasks. In the presented work, we employ a dataset of 444 jazz standard songs harmonizations (melody and chords in lead sheet format) to build and examine the presented methods.

Deep learning architectures have lately become the norm for automatic music generation. Examples of such applications are Hadjeres’ and Pachet’s DeepBach [17], BachBot [18], the BLSTM (bidirectional-LSTM) chord accompaniment system by Lim et al. [19] and Google’s Bach Doodle [20,21]. Although each of these systems operates using different algorithms and techniques, the fundamental principles of music learning and generation underlying them are similar; new harmonic spaces are created by implementing learned representations into probability spaces [22]. While these systems offer promising results, their effectiveness depends on the availability of a large amount of data [23], which is rarely the case in symbolic music applications—certainly so in the case of jazz standards. Furthermore, the mechanisms underlying the harmonization processes are not transparent, not allowing direct explicit control of the output of the model [24], making specific tasks such as blending different harmonic spaces hard to fine tune [23]. On the contrary, the proposed HMM-based approach leads innately to explainable decisions by the model, a fact that has driven gradual improvements, as documented in Section 4.1.

HMMs are very effective for generative music tasks [25], especially when, as in our case, the size of the available dataset is limited. Some innovative approaches in melodic harmonization that employ HMMs are those of Allan and Williams [26], Raphael and Stoddard [27] and Raczynski et al. [28]. In the broader context of music generation, a Markov Model can be used to predict the next notes or chords in a piece of music based on the previous ones by creating a transition matrix. This can be useful for generating music that has some degree of structure and coherence [29] and also can be used to blend different harmonic spaces by combining the chord transition matrices [30]. One limitation of Markov Models, however, is that they are local, meaning that they only consider the immediate surroundings of a given note when making predictions. This can lead to a lack of long-term structure and coherence in the generated music, as the model may not consider the global context of the piece [31]. The problem of locality is a common challenge in music generation using Markov Models [32] and an approach to overcome it is to link the generative process on structure milestones that signify harmonic structural closings or phrase endings, i.e., cadences, using constraints at phrase boundaries [22]. A simple approach for composing melodic harmonizations under this scheme was presented by Kaliakatsos-Papakostas and Cambouropoulos [33], where constraints were added at phrase boundaries, ensuring appropriate cadential schemata at structurally important positions. This method is incorporated in the CHAMELEON melodic harmonization assistant [7,34,35], which is the core system we use in the current work.

Another problem concerning the HMMs models is that the chord transition options can be limited [36]. In the presented case, where we use the harmonic space from only one song (Song B) to re-harmonize a given melody from another one (Song A), this issue is apparent, in the sense that in some cases no chord transitions appearing in Song B may be adequate to harmonize a certain note transition in Song A. To overcome this problem we propose two methods: (a) the use of chord groups, where chords within a group can be used interchangeably, according to the underlying melody, and (b) the use of a “support” harmonic space derived from the overall that represents the collective knowledge of jazz standards, which allows for the resolution of conflicts between the harmony in Song B and the melody in Song A. Both methods aim to provide a musically intuitive way of solving problems when using the proposed re-harmonization approach.

2. Methodology for Jazz Standard Cross-Harmonization

The goal of the presented method is to re-harmonize a jazz standard, Song A, by introducing harmonic components of another jazz standard, Song B; we call this a “cross-harmonization”. This method attempts to implement this re-harmonization by replacing all “replaceable” chord of Song A with a new one that relates to Song B. The chords of Song A that start and end a section are considered “irreplaceable” harmonic constraints; the intention for constraining these chords is to preserve the overall harmonic structure of Song A. The paper at hand presents the concept of cross-harmonization as a methodological component, but the main goal is to present the underlying methods that enable this methodological approach, or other approaches under a similar setup.

The methodology concerns the overall approach to re-harmonizing jazz standards, i.e., Song A with harmonic information from Song B, chord-by-chord, by preserving some harmonic constraints of Song A in the form of chords that start and end sections. Chords are considered in groups and chords within a group can be employed interchangeably, based on their fitness with the underlying melody. Additionally, there is a “background knowledge” harmonic space that comes as “support” when inconsistencies arise between the melodic and harmonic elements of the involved songs. The support space comprises harmonic information from all the available jazz standards in the database. Harmonic information is considered to be reflected by first-order transitions of chord groups, i.e., transition probabilities describe the frequency of occurrence of chord pairs. The method presented herein accommodates the above-mentioned methodology, but it would also facilitate other methodological approaches: not necessarily involving chord-by-chord substitutions (harmonic rhythm transfer); considering sub-spaces for support, based on the styles of

Song A and B; considering higher-order transitions; and considering different grouping approaches per song, among others.

2.1. Data, Overview of Methods and Problem Formulation

The presented methods incorporate (a) a simple method for grouping chords and (b) a method that enables the employment of a support space in a constraint-Hidden Markov Model (cHMM) setup [33]. A “universe” of N chords (HMM states) is considered that comprises 12 roots and 64 jazz chord identified—resulting in $N = 768$ possible chords. These chord types are the ones identified in the commercial app Genius Jam Tracks (<https://geniusjamtracks.com>, last accessed: 19 January 2023). These chords are divided into groups (detailed description in Section 2.2). A set of 444 jazz standards from The Real Book were annotated with chords of this app along with their melodies from real books. Melodies are represented only by their pitch and onset time; no information is retained for their duration or MIDI velocity.

The starting tonality of each piece is considered to be “the tonality” of the entire piece, even though several pieces incorporate clearly tonality changes; all chords within a piece are considered as *scale degrees* of the identified tonality. The decision to consider only one tonality for each piece aims to allow capturing of transposition-related harmonic characteristics within a piece through transitions. For instance, in “So What”, there is a clear transposition from D minor to E $\flat$ minor in Section B—and back to D minor in Section C. Considering this piece to have a single tonality with a D minor label leads to a Markov transition model that has positive probabilities for transitions between chords in the Dm and E $\flat$ m groups. Actually, since all chords in all pieces are considered as scale degrees of the annotated tonality, using “So What” as Song B enables transitions between chords in groups of Im and II $\flat$ m, whenever and if the melody accommodates such a transition. A melodic segment is described by its relative pitch class profile, i.e., a 12-dimensional vector that describes the pitch class profile of all melody notes relative to the tonality of the piece—hereby termed as the “relative pitch class profile” (rPCP) of the melody.

Considering some problem-specific information, the cHMM model requires the following components from Song A (melody and constraints) and Song B (chord transition):

1. $T^{N \times N}$ is the transition probability matrix of Song B. This is built by incrementing the values of an initially zero-valued $N \times N$ that corresponds to the rows of the all the chords that correspond to the group of the first and all the columns of the chords in the group of the second chord, for every chord transition within Song B. Afterwards, all rows are normalized to sum to 1, hence representing transition probability density functions.
2. $A^{12 \times K}$ is the set of melodic segments per chord to be harmonized in Song A, considering K chord positions in Song A. The A matrix incorporates a 12-dimensional column describing the rPCP of the melody notes underlying each of the K chords in Song A.
3. $C^{\{0,1\} \times K}$ is a binary array indicating whether a chord position of Song A is constraint (needs to be substituted by a chord within the same group of the chord already in Song A) or not.

Problem formulation includes the following pieces of background knowledge information:

1. Chords are divided in groups according to their type—details in Section 2.2.
2. $S^{N \times N}$ is the “support” transition probability matrix of all songs in the database (444). This is built by adding the respective T matrix of each piece, weighted by the number of chords within each piece—amounting to the proportional probability contributions of each piece in the final “universal” probability distribution of chord group transitions. This matrix will be employed if the Viterbi algorithm produces no information at a given time step (details in Section 2.3).
3. $M^{N \times 12}$ is the compatibility matrix of each individual chord (not chord group) with an underlying melodic segment, expressed as a 12-dimensional vector (column) that describes the expected melodic rPCP for each chord (row). More details about the construction and involvement of M are given in the following paragraph.

In M , the (initially zero-valued) 12-dimensional vector of each chord is obtained by adding an equal proportion of the rPCP distribution of the chord itself with the distribution of the rPCP of all the underlying melodic segments that this chord has harmonized in all the jazz standards of the database. The values of the 12-tuple that remain zero after this addition, are assigned large negative values (in the presented application, this value is -100). The intuition behind constructing M this way is to make each chord compatible with a melodic segment that either incorporates relative pitch classes of the chord itself, or other relative pitch classes that were present in melodic segments that the chord has harmonized in pieces of the dataset. The strong negative values assigned to the remaining rPCPs penalize decisively the existence of rPCPs that are not relevant (in the two above-mentioned ways) to the chord. This approach aims to eradicate the possibility of assigning chords to melodic segments that incorporate even a single relative pitch class that is irrelevant to them.

The purpose of this matrix is to allow comparisons between the 12-dimensional vector of each chord with the 12-dimensional rPCP of each melodic segment to be harmonised through the formulation of an “observations” matrix, $O^{N \times K} = M^{N \times 12} A^{12 \times K}$. Negative values of O , which are possible because of the large negative values of M , are zeroed out. The comparison is reflected by the magnitude of the values of $O^{N \times K}$ for the respective chord (row) and melodic segment to be harmonized (column). Since these values are obtained through the inner product between the rPCP of the melodic segments to be harmonized (columns of A) and the highly penalizing rPCP expected by each chord (rows of M with large negative values), the large negative will create a negative inner product (or zero value in the final O matrix), even in the case of a single relative pitch class mismatch between the melody and the chord. Therefore, a chord is considered improbable (zero probability), even if a single relative pitch class of the melodic segment to be harmonized has not been encountered in the dataset for the chord under discussion, or if this relative pitch class is not among the relative pitch classes of the chord tones.

It should be noted that chords with no appearance in the dataset (371 chords, or root-type pairs, appearing among the $N = 768$), are also assigned a melody matching profile based only on their relative pitch class profile. This fact enables the employment of such chords, since they can be potentially matched with a melodic segment, while chord grouping can enable transitions to and from these chords, through the groups they belong to. The inclusion of new chords is arguably a positive component of the presented method. The fact that the possibility to add a new chord is based on its rPCP and on the transition probabilities of the group it belongs to is considered a result of common reasoning in musical terms: if the pitches of a new chord fit the melody and the chord group transitions allow its appearance, why not use it?

2.2. Chord Groups

When considering the problem of harmonizing a melody from Song A with the harmony of Song B, one would assume that a broader conceptual consideration of “harmony” would be employed, rather than the strict approach to rely solely on the specific chords that exist in Song B. If chord transitions in the Markov Model that represents Song B incorporate only statistics between chords that are present in Song B, then the scope of the harmony to be employed is rather narrow. The aim of chord grouping is to broaden the harmonic scope by enabling probabilities of transitions between chords of the same group, rather than individual chords themselves.

Chords are divided into six groups according to their type, based on their pitch class content (relative to the root of each chord). The approach to chord grouping employed in this paper is rather basic and possibly naive, since it does not consider the chord functions (e.g., if a chord precedes a tonic chord) and it requires that chords in the same group have the same root. More sophisticated approaches could be explored in future work, e.g., as in [37], but the results presented herein indicate that even the basic approach discussed in this section provides good results. The groups and the “rule” in incorporate a chord type into a specific group are shown in Table 1. For example, the chord with the type $[0, 3, 7, 10]$ (represented as a

pitch class set relative to the root of the chord) belongs to the “Minor” group, since it includes “3” and it does not belong to the “Diminished” group (since it does not include “6”).

Table 1. Chord grouping method and number of chords per group. Chord types are represented by their pitch class set relative to the chord root.

Group	Rule	Num. of Chords
Dominant	$\{4, 10\} \in t \vee (\{4, 8\} \in t \wedge 11 \notin t)$	30
Suspended	$5 \in t \wedge \{3, 4\} \notin t$	6
Diminished	$\{3, 6\} \in t \wedge 10 \notin t$	2
Major	$4 \in t \wedge 10 \notin t$	13
Minor	$3 \in t \wedge t \notin \text{Diminished}$	17
Other	Not in other groups	2

2.3. Supported Constraint Hidden Markov Model

In the typical Hidden Markov Model method, inferring the sequence of a set of known hidden states for each time step t , the Viterbi algorithm is employed, which iteratively accumulates probabilities for each chord being the chord of choice in a matrix Δ and keeps track of the best possible previous chord for each current chord through a matrix of “best previous chord” indexes, ψ . The probabilities in Δ are calculated using the following components:

1. The transition probabilities $T^{N \times N}$.
2. The melody-related fitness of each chord, expressed through the $O^{N \times K}$ matrix.
3. The accumulated Δ value of the best possible previous, based on the transition probability and the melody fitness of the above mentioned matrices.

Specifically, the probability calculated for each chord, j , is the maximum of the element-wise product of each chord being the one in the previous step, times the probability of the examined chord to harmonize the given melody:

$$\Delta[j, t] = \max[(\Delta[:, t-1] \odot T[:, j]) \circ O[j, t]]$$

If T is sparse enough, which is expected in the case for a transition matrix of a single song (Song B), or if there is a misalignment between melody emissions and the learned melodic relative pitch classes for each chord (resulting from the large negative penalty for chord-melody mismatch discussed in Section 2.1), all elements of the current time step column ($\Delta[:, t]$) can be zero.

A common practice in HMM models is to “smoothen” all the incorporated matrices by replacing zero values with a small arbitrary value, for preventing all-zero occurrences in columns of Δ . This allows the continuation of all paths in the trellis diagram until the end of the graph, preventing the Viterbi algorithm from “crushing”, i.e., having all final states of the trellis paths with zero accumulated probability values [38]. Note that zeroing out all the Δ values in a single column will lead to zeroing out all values in all subsequent columns. Smoothing will indeed give a chance to the non-zero component in the computation of $\Delta[j, t]$ to be the deciding factor; i.e., if zero comes as a result of transition impotency, then best melody matching will determine chord selection and vice versa.

Smoothing, however, introduces improbably transitions or melody-to-chord mismatches that are irrelevant to the underlying musical idiom of the jazz standards. The presented method attempts to resolve this issue by providing idiom-relevant solutions to transition or melody incompatibilities. One way this issue is addressed is through chord grouping, as discussed in Section 2.2. This step not only makes transitions more dense, since previously unrelated chords are now connected with non-zero transition probabilities, but also gives more options when it comes to chord-melody matching, since chords with various extensions within a group are simultaneously considered.

Group transitions within a piece, however, might as well be too sparse, since a jazz standard is not necessary to involve transitions between all possible chord groups, let alone

chords in all groups. To this end, a transition “support” ($S[i, j]$) method is proposed that incorporates the transition probabilities of groups of the entire learned idiom. Given enough songs in the idiom, S will be sufficiently denser than T and will lead to fewer occasions where all values in a column of Δ will be zero because of the transition component. To this end, at each time step (t), the current column $\Delta[:, t]$ is examined, and, if it includes all-zero values, it is recomputed with transition probabilities taken from S instead of T . If S , in turn, leads again to an all-zero column, then “smoothed” versions of both S and O are employed, i.e., where all zero values are increased by a small amount (10^{-6}).

The intuition behind using S in “problematic” time steps is to allow the Viterbi algorithm, for those specific time steps, to “borrow” transition ideas from the grouped version of the entire idiom in order to resolve harmonic “dead-ends” that might arise from using only the transitions of a single song. This approach assumes that the “dead-end” is a result of transition probability sparseness, i.e., that all values in a column of Δ are zeroed because of the $T[:, j]$ column in the $\Delta[:, t-1] \odot T[:, j]$ inner product, and not by zero probabilities in melody-chord matching. Similar approaches can be examined for resolving melody mismatches in cases of “dead ends” that occur because of the $O^{N \times K} = M^{N \times 12} A^{12 \times K}$ component. For instance, such an approach could include an iterative relaxation of the melody-per-chord penalties in M , until at least one chord would match the underlying melodic content (in A); this issue needs elaborate examination and is left for future research.

It needs to be noted that using support transition probabilities does not necessarily mean that chord groups that do not belong in Song B are introduced. In fact, in case a single application of support is necessary, this will simply help by indicating how possible it is for groups already within Song B to transit from one to another when considering the transition probabilities of all the jazz standards (not simply the transition probabilities of Song B). This is illustrated in the “Single support” case of the simplified example in Figure 1 (This example refers to “chords” rather than “groups” to avoid complications related to chord-melody matching, when chords are divided in groups. The text, however, refers directly to “groups”, since melody matching is not discussed). If an application of support was leading to a group outside Song B, then it would be necessary to have at least one consecutive application of support transition probabilities to go back to the groups (and the transition space) of Song B. Therefore, it is necessary to have at least two consecutive applications of the support space to allow the employment of at least one single group outside Song B. This possibility is illustrated in the “Double support” case in Figure 1; note that in this example, it is would still be possible to remain within the Song B space in step t , but the illustrated probabilities in this simplified example show how it would be possible to select a group outside Song B.

2.4. Group and Support on Constraints

Chord groups are also effective on locations that are constraint, i.e., starting and ending chords of sections selected by the algorithm need to belong to the group of the initial piece, Song A. On a time step, t , with chord constraint, this is implemented by updating only the rows of $\Delta[:, t]$ that correspond to chords of the group with the chord that exists initially as constraint. If support or smoothing is required in constrained time steps, then those also affect the chords corresponding to the group of the original chord of Song A. Chords involved in section changes are both constraint; it should be noted that, in those cases, all transitions between chords in the groups of the constraints should be possible and, therefore, the values of T and O of the corresponding group chords are smoothed by default—there should not be any check for support. The intuition behind employing groups on positions of chord constraints is to allow the introduction of new chords (of a given group), even in positions of constraints, based on which, “chord variation” is more suitable for the melodic segment under harmonization.

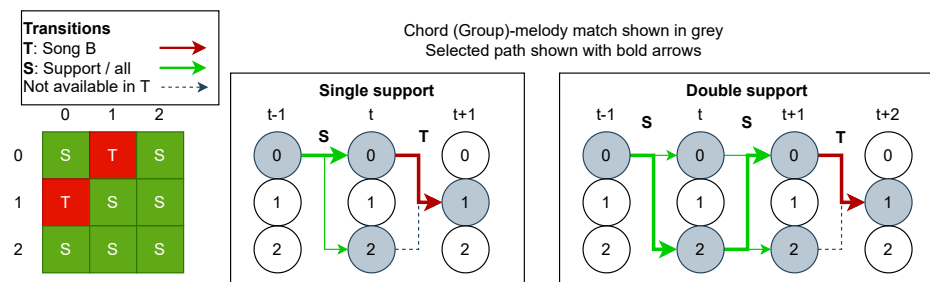


Figure 1. In a simplified example, a “universe” of three chords (equivalent to groups in the presented approach) is considered. *T* includes the transitions of Song B, which incorporate only chords 0 and 1, and transitions are only possible from one to the other (not to themselves). The “support” space includes all three chords, and transitions are possible between all chords (*S* also includes the probabilities of *T*). Some melodic segments are compatible with some chords at time step $t + i$ (as in $O^{N \times K}$); compatible chord-melody matches are indicated in grey, incompatible in white. In case a single support transition probability is used, *S* is only employed to facilitate a transition between chords 0 and 1, which is not possible in *T*; all other paths will be eventually erased by backtracking. If double (or multiple consecutive) supports are employed, then all supports, except from the final in the series, could possibly incorporate chord 2, which is not available in Song B.

3. Results

The presented results aim to describe the behavior of the algorithms, as implemented in the cross-harmonization computational system, in the entire dataset and to examine more closely specific cases in the light of jazz expert knowledge. Specific points of interest include the introduction of support in cross-harmonizations, as well as the introduction of new chords through the employment of chord groups. To this end, a statistical analysis is presented that examines correlations between several components of the application of the method in all possible pairs of Song A–Song B for all pieces in the dataset. Furthermore, expert analysis on selected cross-harmonizations provides a brief overview of the successes and failures of the musical output, while correlations between system-related attributes and expert ratings indicate which probabilistic components may be responsible for good or bad results.

3.1. Statistical Analysis

Cross harmonizations are performed between all song combinations in the dataset, producing $443 \times 443 = 196,249$ harmonizations to be examined. From each harmonization, an explanatory log file is generated in an Excel sheet, which logs information for each chord, including whether a chord is constraint, if it was selected from support or smoothing, if it is a new chord, along with some numerical values concerning transition probability and melody-chord matching, including the melodic rPCP of the segment that each chord harmonizes and the rPCP expected by the chord. Each piece takes part in 443 re-harmonizations as Song A and an equal number as Song B; the aforementioned attributes are collected for each piece, in both cases, acting as Song A and as Song B.

The first goal of this part of the analysis is to examine the overall characteristics of applying support and chord grouping as described above. Table 2 analyses the examined attributes for all pieces in average, as well as for the pieces with higher (“Straight, No Chaser”) and lower (“Falling in Love with Love”) ratios of transitions that came from the support space, over the total number of transitions in the piece. First of all, regarding the average, it appears that support was required in about 42% of the generated chord transitions on average, i.e., an average of 16.12 chords were a result of support transitions on an average number of 38.16 transitions in total, in all cross-harmonization sessions. Additionally, 11.81 of those support transitions come either before, after or over constraint chords. This indicates that constraints are often responsible for the need to employ information from the support space.

Table 2. Statistics of applying support and chord grouping when a piece acts as Song A, i.e., when it provides the melody and the constraints.

Label	Average	Straight, No Chaser (Max. Support)	Falling in Love with Love (Min. Support)
Transitions	38.16	12	44
Constraints	5.32	2	2
Support	16.12	10.77	5.46
Supp. no Constr.	11.81	8.79	4.19
Smoothing	0.17	1	0.00
Smth. no Constr	0.03	1	0
Melody mean	0.26	0.13	0.19
Melody std	0.19	0.11	0.12
New chords	2.22	0.01	0.19

“Straight, No Chaser”, while having only two constraints (beginning and ending chords), needs support for above 10 out of 12 transitions (0.90 ratio) and has the worst melody-to-chord matching with an average of 0.13 (compared to the overall average of 0.26). The increased demand for support in “Straight, No Chaser” is explained by the fact that the melody is highly-chromatic, rendering all 443 pieces (when they act as Song B) incompatible with at least one melodic segment. An exception is also notable with this piece: the average times that smoothing was necessary (where even support failed to provide a solution) is exactly 1. Analysis shows that this happens because of the exceptionally chromatic melodic part in bar 9, which forces the employment of a very specific chord ($\flat$ VI major) to appear, which appears only twice in the support space as the second chord in a transition, with first chords that are incompatible with the melody in the previous bar.

Actually, many of the top-10 pieces in support-per-transition are within the bottom 10 in melody matching. However, this effect diminishes for bottom-10 pieces in support-per-transition, a fact that justifies the overall weak correlation of those two values for all pieces (0.15 in support-per-transition and melody matching over all generated pieces—see row 4, column 2 in Table 3). For instance, the piece with the smaller support-over-transitions value (“Falling in Love with Love”) has indeed a lower-than-average melody matching mean value (0.19). Correlation analysis is conducted on the values of Table 2 for each generated piece, the number of constraints, support and smoothing applications divided by the total number of transitions to obtain results that are relative to the length of each piece. Table 3 shows the results of this correlation analysis, where melody matching is represented by the mean value of melody matches within each piece. The interesting result of this analysis, indicated in bold fonts, is the relatively strong relation between the appearance of new chords with the application of support, over constraints or independently. This fact indicates that, when support is required, the algorithm employs information on chord grouping to provide a solution that best matches the underlying melody with a chord based solely on the rPCP of the latter (since new chords do not include information about the melodic context they expect). It is also interesting that this correlation does not exist for chords that were selected through smoothing.

Table 3. Results of correlation analysis on all generated pieces, for examining relations between constraints, support/smoothing (on constraints or independent), melody matching and the employment of new chords. The following features are examined (a value per feature for each piece): **Constr.**: ratio of constraints over transitions. **Supp./Smth./New**: ratio of times support/smoothing/appearance of new chord was presented over transitions. **-nC** suffix: not coming exactly before, after or over constraint. **Mel**: melody matching value (of $O^{N \times K}$).

	Constr.	Supp.	Supp-nC	Mel.	New	Smth.	Smth-nC
Constr.	1.00	0.13	−0.13	0.00	0.17	0.09	−0.05
Supp.	0.13	1.00	0.96	0.15	0.62	0.28	0.26
Supp-nC	−0.13	0.96	1.00	0.15	0.57	0.25	0.27
Mel.	0.00	0.15	0.15	1.00	0.12	−0.15	−0.18
New	0.17	0.62	0.57	0.12	1.00	0.15	−0.00
Smth.	0.09	0.28	0.25	−0.15	0.15	1.00	0.64
Smth-nC	−0.05	0.26	0.27	−0.18	−0.00	0.64	1.00

3.2. Expert Evaluation and Ratings

One expert on jazz standards (among the authors) was involved in a process of analyzing, commenting in free form and rating a selection of 13 generated harmonizations. The expert would listen to the results and then a digital version of the score was marked. The expert was conducting this process as if he was evaluating students in a harmonization course. The process is explained graphically in Figure 2. The jazz standards that were involved in the re-harmonization sessions were selected based on some exceptional characteristics that they incorporated, according to the expert; the re-harmonizations that are involved in the analysis, along with the ratings in melody, transitions and the average of the two are shown in Table 4. The videos and the log files can be found online (link can be found here: <https://doi.org/10.5281/zenodo.7549632>, last accessed on 19 January 2023).

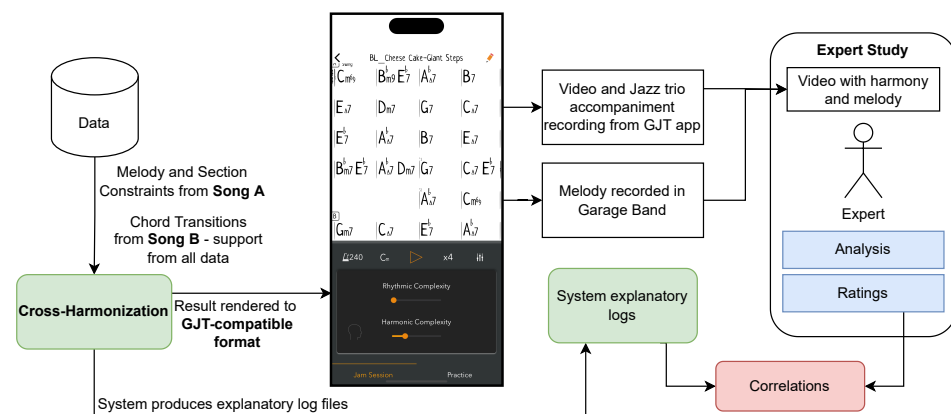


Figure 2. Experimental process pipeline. Pairs of jazz standards are selected and the system employs the presented methods to generate cross-harmonizations between them, along with the respective explanatory log files. The cross-harmonization output is imported to GJT and rendered to video/audio format (one repetition) in jazz trio format (drums, bass and piano). The melody, generated in Garage Band, is imported to the video and the final outcome becomes available to a jazz expert. The expert analyzes the output in free text and the most important comments are gathered. They also rate each harmonization and the correlation is examined between those ratings and several aspects of the generated harmonizations from the log files produced by the system.

Table 4. Re-harmonization sessions and their ratings according to melody matching, selected chords and the average of the two.

Song A	Song B	Melody	Chords	Average
Afro Blue	Giant Steps	9.5	9	9.25
All Of Me	Giant Steps	7.5	6.5	7
All The Things You Are	Darn That Dream	8	8	8
Anthropology	Darn That Dream	9	8.5	8.75
Beautiful Love	Darn That Dream	9.5	8	8.75
Blue Bossa	Giant Steps	6.5	8.5	7.5
Blue Bossa	St. Thomas	6.5	8.5	7.5
C-Jam Blues	Time Remembered	8	4	6
Relaxin' At Camarillo	Time Remembered	7.5	7.5	7.5
St. Thomas	Blue Bossa	10	5.5	7.75
So What	Giant Steps	3	8	5.5
Solar	Giant Steps	1	6	3.5
Solar	Time Remembered	9.5	9.5	9.5

According to the expert analysis, which provided useful insight for the discussion in Section 4.2, the system performed well on several occasions, sometimes even surprisingly well, while, in other cases, some chord decisions were lacking adaptation to the underlying melody and/or harmonic coherence in terms of both chord transitions and individual chord choices. Regarding poor chord choices, an example is the employment of a $m\flat 6$ chord in the first bar of a harmonization; this chord, in the jazz standard idiom, has been employed almost solely as a transitional chord between a $m6$ and a minor triad chord. Regarding melody mismatches, the system, on some occasions, harmonized melody pitches that are included in the “avoid notes” of jazz standard terminology, e.g., a major third over a minor chord, or the 11th over a major chord. Erroneous note harmonization was performed not only in overall bad harmonization results, e.g., in the “Solar–Giant Steps” cross-harmonization shown Figure 3a, but also in segments of good overall harmonizations, where one or two chords were not producing satisfactory matching with the underlying melody, e.g., in the “Solar–Time Remembered” cross-harmonization. In the latter case, the harmonization was overall satisfactory and the broader harmonic concept of Song B (Time Remembered) was indeed passed in the cross-harmonization: no dominant chords were employed, except for the last chord, which is a constraint chord of Song A.

Regarding the ratings, they were based, as much as possible, on an “academic” point of view, which was pursued by instructing the music expert to rate as if they were rating re-harmonizations of students. The ratings (on a scale of 0–10) show that the algorithms are, in general, capable of generating re-harmonizations that are rated high, i.e., above 8 (5/13), while all except three harmonizations are rated above 7 (10/13). There are three cases with poor rating, namely 6, 5.5 and 3.5.

The ratings of each re-harmonization are examined in terms of their correlation with some features obtained from the log files. Specifically, the correlation is studied between the rating and the accumulated number of constraints per transition, supported transitions per transition, new chords per transition, transition probabilities and melody matching in each re-harmonization. The results are shown in Table 5. Even though the re-harmonizations are very few (13) and, additionally, there are no high-enough values to show clear correlations, the indicated weak relations provide interesting insight that can be valuable for future studies that focus more on the qualitative, subjective evaluation of the methods.

(a) Solar–Giant Steps

(b) Solar–Time Remembered

Figure 3. Two example cross-harmonizations with “Solar” as Song A and (a) Giant Steps and (b) Time Remembered as Song B. Melody mismatches are indicated. In (a), melody mismatches are very often, in contrast to (b).

Table 5. Correlation analysis between ratings and re-harmonization features. Noteworthy correlation values are indicated in bold.

	Melody	Chords	Average
Constraints	−0.42	0.11	−0.29
Support	−0.51	−0.12	−0.47
New chords	0.30	0.43	0.45
Transition prob.	−0.12	0.29	0.04
Melody matching	0.42	−0.51	0.09

Table 5 indicates that the average rating is correlated negatively with the number of supported transitions, while it is positively correlated with the appearance of new chords, i.e., better rates correspond to less supported transitions but more new chords. However, in the entire set of all 196,249 re-harmonizations, the number of supported transitions is correlated with the number of new chords. It happens that the 13 re-harmonization tasks that are examined exhibit no correlation between these two features (i.e., correlation between number of supported chords per transition and new chords per transition is 0.08). The negative correlation component, regarding the supported transitions, is related with melody rating, i.e., the fewer

the supported chords, the higher the melody rating. The positive component, regarding the number of appearances of new chords, is related to transition ratings, i.e., the more new chords, the higher the transitions rating. Therefore, there appears to be more complex relations between good melody matching, good chord transitions, supported transitions and new chords that needs to be further examined in future research.

Another aspect of the correlation analysis that needs to be noted is the fact that the melody matching measured in the log file (values of $O^{N \times K}$) correlates positively with the melody rating (which is expected), but is negatively correlated with chord transition rating. This means that better ratings in chord transitions are a result of costly compromises regarding melody matching. Again, this result provides an interesting topic to be examined more thoroughly in a study that focuses on the subjective impact of the proposed algorithms.

4. Method Evolution and Future Improvements

The presented approach and methods constitute a relatively robust version of the algorithms that resulted from studying the behavior of well-established HMM-based methods within the context of cross-harmonization. This section presents the steps that led to the current version and some next steps that we assume will lead to improved versions. Additionally, some possible practical applications of the overall cross-harmonization methodology are discussed in potential real-world settings.

4.1. Method Development Milestones

This section highlights the basic steps that were followed and led to the development of the presented version of the method. These steps are divided into three “milestone” versions of the algorithms that are briefly outlined. There are two goals in presenting those milestones: (a) to clarify in isolation the improvement of each component that was added or modified in incremental fashion in the proposed method and (b) to discuss what has been tested in previous versions and how and why it did not work.

Milestone 1. The first attempt was based on a straightforward idea: to create a weighted average between the transition matrices of Song B and of all the songs of the entire idiom (support matrix), e.g., create a transition matrix as the sum $0.7 \cdot T + 0.3 \cdot S$. This idea involved smoothening zero entries for avoiding zero column entries in Δ and did not incorporate any penalty for chord-melody mismatch; the chord with the strongest match with the melody was receiving a higher probability, regardless of how many melodic notes it was missing. The problem with this version was that even the slightest contribution of S in the weighted sum led ultimately to the prevalence of S , which was “saturating” the trellis diagram, almost regardless of the selection of Song B, leading to “run-of-the-mill” re-harmonizations that reflected generic attributes of the jazz standards. This behaviour was occurring because even the slightest incompatibility between the melody or constraints in Song A and the chord transitions in Song B was sidetracking the algorithm to select songs that ultimately belonged only to S .

Milestone 2. The second version did not incorporate smoothening of the involved matrices and was employing support solely in the case of a zero column in Δ , as presented in Section 2.3. This approach was not sidetracking all the decisions of the algorithm, since, at any given time step, only T was examined first and S was employed only if all possibilities for transition in T were zero. With this approach, at any given time step, only the trellis paths that were providing at least the slightest possibility for chords in Song B (expressed by transitions in T) were first examined, disregarding S and disallowing it to “saturate” the trellis diagram. The problem with this version was that chord selection was very restrictive to the exact types in Song B. Since no chord-melody mismatch penalty was yet introduced at this point, the exact chords of Song B would somehow fit in melodic segments that involved at least one matching relative pitch class. However, when music experts were presented with the results of this version, they were constantly identifying “missed opportunities” for employing chords of the same overall quality (e.g., major, minor,

dominant, etc.), but with specific extensions and alterations that were more suitable to the underlying melody.

Milestone 3. The third version, the one before the version presented in this paper, integrated chord grouping as discussed in Section 2.2. This rudimentary approach to chord grouping enabled great improvement, according to music experts, since the basic harmonic concepts of Song B were becoming present, even if they were expressed with chord types that were not exactly the same as in Song B. Substituting chord types was considered, in fact, by music experts as a sign of “adaptability” of the algorithm to the demands of the melodic conditions in Song A. The problem with this version was that the algorithm was too forgiving when it came to melody-chord mismatches. The current version, presented in Section 2, added a large penalty for even a single melody-chord mismatch, which improved the results significantly. There are still identified aspects that can be improved, but the current version presents an opportunity to focus the study clearly on the effects of grouping and support utilization. Some possible pathways for improvement are presented below, but their effect will need to be carefully examined, in isolation, in future studies.

4.2. Possible Pathways to Improvement

Both chord grouping and transition support provide a framework that can integrate several further components. The first identified weakness of the current implementation concerns melody matching, which appears to be too “stiff” in some cases and too “forgiving” in others. This is possibly due to the rudimentary method for melody representation. To this end, melody representation, even in the current form of the 12-dimensional rPCP vector, needs to incorporate more musically meaningful aspects, e.g., duration, weak-versus-strong metrical position, etc. Additionally, a more refined method for chord tolerance to melodic mismatches can be introduced, possibly involving ad hoc rPCPs that can and cannot be acceptable for each chord type. E.g., all major group chords, regardless of their specific type and what they have harmonized in the dataset, could accept a melody note that corresponds to the 6th of the chord, but could not accept a minor third note. In the current form, such simple musical considerations might be violated due to rare exceptions in data that do not apply to generic music knowledge. Additionally, characteristic melody matching information individually from Song A and/or Song B could be involved, instead of or in combination with melody matching information from the overall space of jazz standards, as in the current version. Such ideas, however, need to be tested in a study that provides results focusing on this specific aspect.

Apart from melodic representation issues discussed above, chord representation may also be improved in more musically meaningful ways. For instance, chord grouping could be more refined, incorporating the functional and/or structural role of each chord, or the mode of the piece tonality (e.g., requiring transposition to the relative minor or major in Song B, to match the mode in the tonality of Song A. For instance, chord grouping could involve the function of each chord within the involved songs, or in the overall space, along with the metrical position of the chord, i.e., if it is placed on a weak or a strong beat. Such characteristics do not necessarily need to be integrated in the transition matrix as the current version of chord grouping, but can be added as a separate layer of probabilities during the computation of Δ at each time step, i.e., instead of having two layers of probabilities at each time (transitions and melody matching), more probabilities per state/chord could be integrated directly as a product of probabilities, based, e.g., on the metrical position of the time step and the learned role of each chord in this metrical position.

Another issue that relates to the overall statistical learning approach is the inherent attribute of such methods to focus on statistical “normalities”, disregarding idiosyncratic aspects of data that are considered “anomalies”, which, however, might be valuable for identifying useful characteristics in specific data points. For instance, in “So What”, the most salient harmonic phenomenon is the semitone step in the entire tonality in Section B, which is, however, poorly reflected by the transition probabilities in the model of the piece that gives more weight to repeated transitions. Such idiosyncratic elements could be highlighted

by introducing a “salience” component, e.g., as developed in [39], either in the formulation of the transition probabilities or as a separate layer in the formulation of Δ .

5. Conclusions

This paper presents a new methodology for creating novel melodic harmonizations of jazz standards through cross-harmonization, i.e., harmonization of the melody and some chord constraints of Song A, with the chords and transitions of Song B. While developing the basic Hidden Markov Model in collaboration with jazz experts for achieving this task, two new methods were created, which function on top of the Viterbi algorithm: the incorporation of chord groups (rather than individual chords as states) and the introduction of a “supporting” chord transition space, trained on the entire dataset of jazz standards, at the exact time steps where incompatibilities between the melody or the constraints of Song A and the chord groups and transitions of Song B are encountered. Results were presented on a dataset of 444 chord-melody annotated jazz standards, providing an understanding on how often the “supporting” space was employed and how often that led to the introduction of new chords that were never used in other jazz standards of the dataset. Expert analysis showed that the main problem, in some cases, is that there are still significant chord-melody mismatches. Possible solutions are discussed, along with a discussion about other possibilities for improvement. From the results presented herein, it can be concluded that the presented approach has the potential to lead to new interesting applications pertaining to the field of computational creativity in music, while it opens up new research directions both for improvements of the proposed methods and for the development of other methods that facilitate cross-harmonization.

Two limitations of the presented study are the small dataset that is currently available and the fact that the proposed approach has not been assessed within the context of practical applications, such as music learning. Except for the method-related issues discussed in Section 4.2, application and evaluation issues should be additionally addressed. The inclusion of more jazz standards will provide further insight into how the support space can introduce new solutions, while it will allow the examination of support “subspaces” that do not include the entire dataset, but a smaller subset of the dataset the Song B belongs to. Evaluation needs to address issues related to the feasibility and the usability of the presented approach in specific application scenarios. The current results aim to shed some light on how the methods behave, mainly based on some quantifiable criteria, but there exist concrete real-world scenarios for which the effectiveness of the proposed approach needs to be assessed.

The presented re-harmonization methodology allows intertwining of a given harmonic structure, i.e., that of Song B, with a combined melodic-harmonic context, i.e., that of Song A. The context of Song A integrates melodic and harmonic components, both in the form of the constraint chords (beginning and ending of sections), as well as in the form of the harmony implied by the melody under consideration. The results indicate that interesting harmonic phenomena can emerge for some combinations of Song A and Song B, although there are cases in which algorithm decisions may lead to re-harmonizations that are not favorable for the idiom of jazz standards as it stands. In such cases, applications are sought in settings that can either have a higher degree of tolerance to errors, or that have exploratory settings, in which errors may be of interest for developing new musical idioms. In live performances, musicians will often start performing one piece without concluding the previous one. This is usually done by developing the melodic theme of the second piece within the duration of the first piece. The methodology presented here can offer ideas for such seamless transitions between two pieces of music, which, in an analogy to the signal processing terminology, may be considered to provide some form of ‘harmonic morphing’. This paradigm of temporary morphing or permanent blending of two pieces of music can be useful in the context of music pedagogy, as well as under a more controversial perspective, for example, in human-machine performances (a.k.a. computer accompaniment).

In music learning and teaching, practicing the same piece of music repeatedly becomes tedious and uninteresting both for students as well as for teachers. The proposed application allows the transformation of the piece, while, at the same time, encouraging the performer to develop new ideas borrowed or stolen from another piece, therefore enhancing student engagement and enriching the learning experience. Such an application may be especially useful in online learning platforms, through which teacher–student interactions essentially employ virtual tools to develop the educational practices. In such setups, unsuccessful re-harmonizations may be rated by users (students and teachers), therefore further informing transition matrices to guide the Viterbi algorithm to increasingly provide more successful harmonizations. The methods presented in this paper will be deployed and assessed in the context of the pilot experiments of the MusiCoLab project (<https://musicolab.hmu.gr/wordpress/>, last accessed on 19 January 2023) for online music education [40].

Furthermore, real-time re-harmonisations may be realized in the domain of human-machine improvisation. Computer accompaniment systems have a history of several decades [41,42]. They aim at developing artificial performers able to collaborate with humans in learning, practicing and enjoying music performance, when it is not possible for the latter to meet their collaborating partners [43,44]. A creative harmonisation agent, augmented with machine listening capabilities, i.e., capable of following the performance of a human in terms of timing and dynamics, can be used to provide an indefinite number of re-harmonisations of a given piece. These re-harmonisations may be used as backing tracks for the human performer, derived from numerous jazz standards, therefore challenging the performer to indefinitely follow new ideas built on prior background knowledge.

Author Contributions: M.K.-P. was involved in developing and implementing the methods, writing and coordinating the writing of parts the paper. The overall methodology and the general idea of the methods was conceptualized by E.C., K.V. was involved in parts of the implementation and the literature review. L.P. provided musical expertise that shaped algorithm-related decisions. C.A. contributed by providing a framework for real-world potential applications. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE. Project Acronym: MusiCoLab, Project Code: T2EDK-00353.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data will be uploaded in Zenodo upon acceptance.

Acknowledgments: The authors would also like to acknowledge the valuable comments and work of Costas Tsougras, Konstantinos Giannos and Eirini Gkougkoustamou during the development of the presented methods.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Fauconnier, G.; Turner, M. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*, reprint ed.; Basic Books: New York, NY, USA, 2003.
2. Zbikowski, L.M. *Conceptualizing Music: Cognitive Structure, Theory, and Analysis*; Oxford University Press: Oxford, UK, 2002.
3. Zbikowski, L.M. Metaphor and music. In *The Cambridge Handbook of Metaphor and Thought*; Cambridge University Press: Cambridge, UK, 2008; pp. 502–524.
4. Zbikowski, L.M. Conceptual blending, creativity, and music. *Music. Sci.* **2018**, *22*, 6–23. [[CrossRef](#)]
5. Goguen, J. Mathematical models of cognitive space and time. In *Reasoning and Cognition*; Andler, D., Ogawa, Y., Okada, M., Watanabe, S., Eds.; Interdisciplinary Conference Series on Reasoning Studies; Keio University Press: Tokyo, Japan, 2006; Volume 2.
6. Goguen, J.; Harrell, D.F. Style: A computational and conceptual blending-based approach. In *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*; Argamon, S., Dubnov, S., Eds.; Springer: Berlin, Germany, 2010; pp. 147–170.

7. Kaliakatsos-Papakostas, M.A.; de Queiroz, M.G.; Tsougras, C.; Cambouropoulos, E. Conceptual Blending of Harmonic Spaces for Creative Melodic Harmonisation. *J. New Music Res.* **2017**, *46*, 305–328. [\[CrossRef\]](#)
8. Fauconnier, G. Conceptual blending and analogy. *Analog. Mind Perspect. Cogn. Sci.* **2001**, *255*, 286.
9. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **1967**, *13*, 260–269. [\[CrossRef\]](#)
10. Forney, G. The viterbi algorithm. *Proc. IEEE* **1973**, *61*, 268–278. [\[CrossRef\]](#)
11. Gillick, J.; Tang, K.; Keller, R.M. Machine learning of jazz grammars. *Comput. Music J.* **2010**, *34*, 56–66. [\[CrossRef\]](#)
12. Dannenberg, R.B. Dynamic programming for interactive music systems. In *Readings in Music and Artificial Intelligence*; Miranda, E.R., Ed.; Harwood Academic Publishers: Amsterdam, The Netherlands, 2000; pp. 189–206.
13. Hori, T.; Nakamura, K.; Sagayama, S. Jazz piano trio synthesizing system based on HMM and DNN. In Proceedings of the 14th Sound and Music Computing Conference 2017, SMC 2017, Espoo, Finland, 5–8 July 2017; Lokki, T., Patynen, J., Valimäki, V., Eds.; Aalto University: Espoo, Finland, 2019; pp. 153–158.
14. Hung, H.T.; Wang, C.Y.; Yang, Y.H.; Wang, H.M. Improving automatic jazz melody generation by transfer learning techniques. In Proceedings of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 18–21 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 339–346.
15. Keller, R.; Morrison, D. A grammatical approach to automatic improvisation. In Proceedings of the 4th Sound and Music Computing Conference, SMC, Lefkada, Greece, 11–13 July 2007.
16. Wiggins, G.; Smaill, A. Musical Knowledge: What can Artificial Intelligence bring to the musician? In *Readings in Music and Artificial Intelligence*; Routledge: London, UK, 2013; pp. 39–56.
17. Hadjeres, G.; Pachet, F.; Nielsen, F. Deepbach: A steerable model for bach chorales generation. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1362–1371.
18. Liang, F.T.; Gotham, M.; Johnson, M.; Shotton, J. Automatic Stylistic Composition of Bach Chorales with Deep LSTM. In Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, 23–27 October 2017; pp. 449–456.
19. Lim, H.; Rhyu, S.; Lee, K. Chord generation from symbolic melody using BLSTM networks. *arXiv* **2017**, arXiv:1712.01011.
20. Huang, C.A.; Hawthorne, C.; Roberts, A.; Dinculescu, M.; Wexler, J.; Hong, L.; Howcroft, J. The Bach Doodle: Approachable music composition with machine learning at scale. *arXiv* **2019**, arXiv:1907.06637.
21. Huang, C.A.; Cooijmans, T.; Roberts, A.; Courville, A.C.; Eck, D. Counterpoint by Convolution. *arXiv* **2019**, arXiv:1903.07227.
22. Cambouropoulos, E.; Kaliakatsos-Papakostas, M. Cognitive musicology and Artificial Intelligence: Harmonic analysis, learning, and generation. In *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*; Miranda, E.R., Ed.; Springer International Publishing: Cham, Switzerland, 2021; pp. 263–281. [\[CrossRef\]](#)
23. Garnelo, M.; Shanahan, M. Reconciling deep learning with symbolic artificial intelligence: Representing objects and relations. *Curr. Opin. Behav. Sci.* **2019**, *29*, 17–23. [\[CrossRef\]](#)
24. Marcus, G. Deep learning: A critical appraisal. *arXiv* **2018**, arXiv:1801.00631.
25. Ames, C. The Markov process as a compositional model: A survey and tutorial. *Leonardo* **1989**, *22*, 175–187. [\[CrossRef\]](#)
26. Allan, M.; Williams, C. Harmonising chorales by probabilistic inference. In *Advances in Neural Information Processing Systems*; Saul, L., Weiss, Y., Bottou, L., Eds.; MIT Press: Cambridge, MA, USA, 2004; Volume 17.
27. Raphael, C.; Stoddard, J. Functional harmonic analysis using probabilistic models. *Comput. Music J.* **2004**, *28*, 45–52. [\[CrossRef\]](#)
28. Raczynski, F.; Fukayama, E. and Vincent]. Raczynski, S.A.; Fukayama, S.; Vincent, E. Melody Harmonization With Interpolated Probabilistic Models. *J. New Music Res.* **2013**, *42*, 223–235. [\[CrossRef\]](#)
29. Koops, H.V.; Magalhães, J.P.; de Haas, W.B. A Functional Approach to Automatic Melody Harmonisation. In Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling and Design, FARM '13, Boston, MA, USA, 25–27 September 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 47–58. [\[CrossRef\]](#)
30. Yi, L.; Goldsmith, J. Automatic Generation of Four-part Harmony. In Proceedings of the Fifth UAI Conference on Bayesian Modeling Applications Workshop, Vancouver, BC, Canada, 19 July 2007; Volume 268, pp. 81–86.
31. Dixon, S.; Mauch, M.; Anglade, A. Probabilistic and logic-based modelling of harmony. In *Exploring Music Contents*; Ystad, S., Aramaki, M., Kronland-Martinet, R., Jensen, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–19.
32. Miranda, E. *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021.
33. Kaliakatsos-Papakostas, M.; Cambouropoulos, E. Probabilistic harmonisation with fixed intermediate chord constraints. In Proceedings of the Joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC), ICMC–SMC 2014, Athens, Greece, 14–20 September 2014.
34. Kaliakatsos-Papakostas, M.; Makris, D.; Tsougras, C.; Cambouropoulos, E. Learning and Creating Novel Harmonies in Diverse Musical Idioms: An Adaptive Modular Melodic Harmonisation System. *J. Creat. Music Syst.* **2016**, *1*. [\[CrossRef\]](#)
35. Zacharakis, A.; Kaliakatsos-Papakostas, M.; Kalaitzidou, S.; Cambouropoulos, E. Evaluating Human-Computer Co-creative Processes in Music: A Case Study on the CHAMELEON Melodic Harmonizer. *Front. Psychol.* **2021**, *12*, 603752. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Paiement, J.F.; Eck, D.; Bengio, S. A Probabilistic Model for Chord Progressions. In Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR), London, UK, 11–15 September 2005; pp. 312–319.

37. Kaliakatsos-Papakostas, M.; Zacharakis, A.; Tsougras, C.; Cambouropoulos, E. Evaluating the General Chord Type representation in tonal music and organising GCT chord labels in functional chord categories. In Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2015), Malaga, Spain, 28–30 October 2015.
38. Chen, S.F.; Goodman, J. An empirical study of smoothing techniques for language modeling. *Comput. Speech Lang.* **1999**, *13*, 359–394. [[CrossRef](#)]
39. Kaliakatsos-Papakostas, M.; Cambouropoulos, E. Conceptual blending of high-level features and data-driven salience computation in melodic generation. *Cogn. Syst. Res.* **2019**, *58*, 55–70. [[CrossRef](#)]
40. Alexandraki, C.; Akoumianakis, D.; Kalochristianakis, M.; Zervas, P.; Maximos, K.P.; Cambouropoulos, E. MusiCoLab: Towards a modular architecture for collaborative music learning. In Proceedings of the Web Audio Conference, Cannes, France, 6–8 July 2022.
41. Dannenberg, R.B. Human computer music performance. In *Dagstuhl Follow-Ups*; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik: Wadern, Germany, 2012; Volume 3.
42. Dannenberg, R.B. A Vision of Creative Computation in Music Performance. In Proceedings of the Second International Computer Music Conference, Singapore, 29 September–4 October 2003.
43. Raphael, C. Orchestral Musical Accompaniment from Synthesized Audio. In Proceedings of the International Conference on Computational Creativity, Mexico City, Mexico, 27–29 April 2011; pp. 84–89.
44. Raphael, C. Music plus one: A system for flexible and expressive musical accompaniment. In Proceedings of the International Computer Music Conference, Havana, Cuba, 17–2 September 2001; pp. 159–162.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.