CONSTRUCTING COMPOSITE FEATURES FOR INTERPRETABLE MUSIC-TAGGING

Chenhao Xue[1] *Weitao Hu*[2] *Joyraj Chakraborty*[1] *Zhijin Guo*[1]
Kang Li[1] *Tianyu Shi*[3] *Martin Reed*[4] *Nikolaos Thomos*[4]

[1]University of Oxford [2]Independent Researcher [3]University of Toronto [4]University of Essex

ABSTRACT

Combining multiple audio features can improve the performance of music tagging, but common deep learning-based feature fusion methods often lack interpretability. To address this problem, we propose a Genetic Programming (GP) pipeline that automatically evolves composite features by mathematically combining base music features, thereby capturing synergistic interactions while preserving interpretability. This approach provides representational benefits similar to deep feature fusion without sacrificing interpretability. Experiments on the MTG-Jamendo and GTZAN datasets demonstrate consistent improvements compared to state-of-the-art systems across base feature sets at different abstraction levels. It should be noted that most of the performance gains are noticed within the first few hundred GP evaluations, indicating that effective feature combinations can be identified under modest search budgets. The top evolved expressions include linear, nonlinear, and conditional forms, with various low-complexity solutions at top performance aligned with parsimony pressure to prefer simpler expressions. Analyzing these composite features further reveals which interactions and transformations tend to be beneficial for tagging, offering insights that remain opaque in black-box deep models.

***Index Terms*—** Music tagging; Feature construction; Genetic programming; Interpretability; Music information retrieval.

1. INTRODUCTION

Music audio tagging concerns automatically assigning descriptive labels or "tags" (e.g., mood, theme, instrument) to music tracks based on their audio content. This is a fundamental problem in music information retrieval (MIR) because accurate tags enable the efficient organization and retrieval of large music collections [1]. For decades, automatic tagging has been approached via handcrafted feature extraction followed by traditional machine-learning classifiers [2, 3, 4, 5]. These engineered features were designed to capture low- and mid-level acoustically and musically significant characteristics (e.g., MFCC for timbre or chroma features for tonality), which made them inherently interpretable as they correspond to perceptual aspects of music. In recent years, there has been a shift towards minimal feature engineering and end-to-end deep learning models that learn features directly from audio [6, 7, 8]. These methods significantly enhance the accuracy of music tagging models with overwhelming parameters, vast labeled data, and synergy modelling of individual music features (closer to human perception [9, 10]) with explicit or implicit complex feature fusion [1, 11, 12].

While deep-learning models achieve state-of-the-art tagging performance, their opacity remains a concern, as they cannot easily explain why a certain tag was assigned. This is exacerbated in tasks with subjective or ambiguous ground truth [13]. The subjective nature of tags like "happy" or "aggressive" means that even ground truth annotations can vary between annotators [13, 14]. A black-box model could latch onto spurious correlations. Without interpretability, neither developers nor users can tell if the model is leveraging unwarranted biases (e.g., associating "sadness" only with slow tempo) [15]. Indeed, a lack of transparency in the tagging decisions makes it difficult to detect dataset biases or flaws in model reasoning. This is critical in music machine learning (ML) where models often train on limited genres or cultures, and hidden biases may lead to poor generalization or unfair outcomes [16].

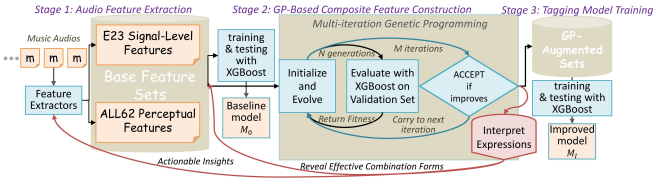

Fig. 1. GP pipeline for Composite Feature and Model Construction.

These practical considerations call for interpretable music audio tagging methods that could approach some advantages of deep learning models, such as the modelling of synergetic effects of music features, which has received limited study in music tasks [17, 18]. In this paper, we introduce a method to automatically construct interpretable composite features from individual base features for music tagging using Genetic Programming (GP), which is a form of symbolic regression [19, 20]. Our specific contributions are:

- we propose a Genetic Programming pipeline that constructs interpretable composite features by combining base music features to improve tagging and reveal interaction insights;
- we demonstrate that the GP-augmented feature sets consistently improve tagging accuracy across two datasets and different types of base features, with gains achieved under modest evaluation budgets;
- we analyze the resulting symbolic expressions to identify effective feature interactions and transformations, offering interpretability and insight not available in black-box models.

The GP-constructed composite features, expressions, and code for this paper are made available on GitHub[1].

2. METHODS

Although deep learning-based music tagging methods achieve state-of-the-art music-tagging performance, their feature fusion remains

The authors thank Puyu Wang for providing music theory validation of the evolved GP expressions.

[1]https://github.com/ChenHX111/GP-Music-Tagging

opaque, and hence, making it difficult to understand why a tag has been allocated to a piece of music. Traditional handcrafted approaches provide interpretability through feature importance, but cannot systematically discover complex feature combinations at scale. To address this challenge, we propose a GP pipeline that evolves interpretable mathematical combination expressions, providing explicit symbolic representations of feature interactions while enabling automated discovery of effective combinations. As shown in Figure 1, the pipeline consists of three stages: (1) Audio Feature Extraction, (2) GP-Based Composite Feature Construction, and (3) Tagging Model Training.

2.1. Audio Feature Extraction

We employ two feature sets at different abstraction levels, following established music information retrieval (MIR) practices that combine signal descriptors with perceptual knowledge [5]. Specifically,

Signal-level Features (E23): We extract 23-dimensional audio descriptors using the Essentia library's music feature extractor [21]. This base feature set captures signal-level characteristics (e.g., Loudness, BPM, Onset Rate, Zero Crossing Rate, Dynamic Complexity, Pitch Salience, and Spectral-Centroid);

Low and Mid-Level Perceptual Features (ALL62): We use a 62-dimensional feature set from Lyberatos et al.'s study [5]. This set includes: (a) the E23 features described above, (b) 32 ontology-grounded harmonic-function features computed using the Omnizart chord recognition Python library [22] mapped to Functional Harmony Ontology classes with normalized n-gram frequencies queried via SPARQL [23], and (c) 7 perceptual features estimated through multi-output regression using a VGG-style Convolutional Neural Networks (CNN) processing Mel-Frequency Cepstral Coefficients (MFCC) segments derived from 15-sec clips.

Before further processing, all features are scaled to have zero mean and unit variance. These base feature sets are input for GP-based composite feature construction.

2.2. GP-Based Composite Feature Construction

Let us define base features $\mathbf{X} = \{x_1, \ldots, x_n\}$ and target labels $\mathbf{y}$. We first formulate the GP composite feature construction as an optimization problem:

$$\max_{f_1,\ldots,f_M \in \mathcal{F}} P(\mathbf{X} \cup \{f_1(\mathbf{X}), \ldots, f_M(\mathbf{X})\}, \mathbf{y}) \; - \; \lambda \sum_{i=1}^{M} \ell(f_i)$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ are composite features from function space $\mathcal{F}$ defined by GP operations $\mathcal{O}$. $P(\cdot, \mathbf{y})$ represents the tagging performance metric. $\ell(f_i)$ is the expression size (number of nodes, i.e., operators and terminals) in the expression tree of f_i (see Figure 2). Our GP system evolves interpretable mathematical expressions that solve this optimization problem at a scale, generating human-readable expressions that reveal feature interactions. Once evolved, these expressions require only basic computations for new instances, concentrating computational cost at training time rather than deployment.

We evolve up to M composite features in M iterations using base features as primitives. Each iteration consists of a full GP run. In our framework, each GP *individual* encodes a scalar composite feature as a rooted expression tree, constructed from standardized base features together with ephemeral constants sampled from $c \sim \mathcal{U}[-2, 2]$. The operation set $\mathcal{O}$ includes arithmetic and protected numeric operators (e.g., `log`, `sqrt`, `div`, `inv`) for linear/nonlinear combination; trigonometric and hyperbolic functions for periodic patterns; `min`/`max`; neural-style activations (e.g., `sigmoid`, `RELU`, `LRELU`, `swish`); and a ternary `if_then` for piecewise dependencies, with closure ensuring real-valued outputs. Initialization uses ramped half-and-half (depth in [1,3]). Population evolution uses one-point subtree crossover (with rate 0.8, illustrated in Figure 2); uniform subtree mutation (rate 0.1) replacing a randomly selected subtree with a full tree of depth in [0,2]; tournament selection (size 3); and a static height limit of 6 to control bloat. Numerical robustness is enforced via protected primitives, fitness penalties for invalid values (`NaN`/∞), sanitization via `nan_to_num`, and standardization of each candidate feature before evaluation.

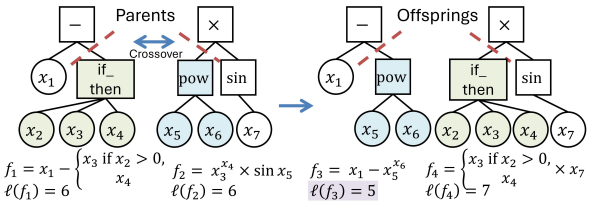

Fig. 2. GP crossover operation for composite feature construction

Table 1. GP-Augmented vs. Baseline Performance for deployment models

Method	MTG-Jamendo (AUC)		GTZAN (ACC)	
	Score [95% CI]	$+\Delta$	Score [95% CI]	$+\Delta$
ALL62 [5, 26]	0.727	–	0.765	–
ALL62 + **GP100**	0.729 [0.724–0.733]	0.002	0.800 [0.760–0.845]	0.035
ALL62 + **GP500**	0.730 [0.724–0.736]	0.003	0.805 [0.760–0.850]	0.040
E23 [5, 26]	0.719	–	0.740	–
E23 + **GP100**	0.722 [0.716–0.728]	0.003	0.785 [0.730–0.830]	0.045
E23 + **GP500**	0.724 [0.717–0.731]	0.005	0.790 [0.735–0.840]	0.050

The fitness function evaluates candidates by augmenting the feature set and measuring XGBoost [24] held-out validation set performance, thereby directly optimizing for predictive utility. Following parsimony pressure [25], we apply a complexity penalty ($\lambda = 0.01$ per node found by experimentation) to promote interpretability by favoring simpler expressions over complex ones (e.g., f_3 over f_4 in Figure 2), preventing bloat and maintaining transparency. The evolved expressions provide interpretability through two mechanisms: first, each expression explicitly shows how base features combine at scale revealing various interactions unavailable in both black-box models [11, 12] and beyond systematic exploration capabilities of manual feature engineering [5, 26], and second, iterative single-feature addition directly assesses each composite feature's contribution.

2.3. Tagging Model Training

We employ XGBoost [24] as our tagging classifier due to its strong performance on tabular data and robustness to mixed feature types. MTG-Jamendo mood/theme tagging is treated as multiple binary classification tasks (multilabel) [27], while GTZAN genre tagging uses multiclass classification [28]. We evaluate using ROC-AUC for MTG-Jamendo and accuracy for GTZAN, consistent with established interpretable [5, 26] and deep learning methods [29, 30].

3. EXPERIMENTAL RESULTS

3.1. Dataset and Setup

Our GP implementation uses populations of 100 and 500 individuals via DEAP [31], with termination after 50 generations or early stopping (stagnation: if no improvement for 15 generations; con-

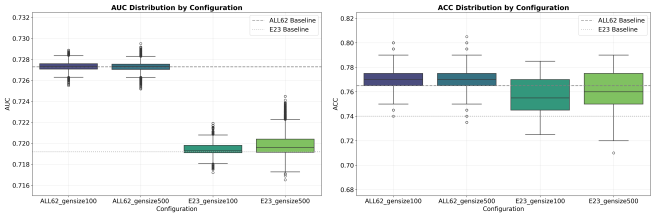

Fig. 3. MTG-Jamendo AUC Distribution of all augmented sets

Fig. 4. GTZAN Accuracy Distribution of all augmented sets

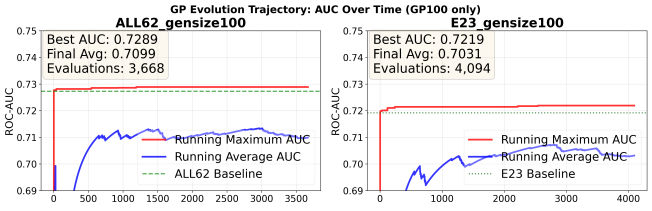

Fig. 5. MTG-Jamendo Improvement Trajectory

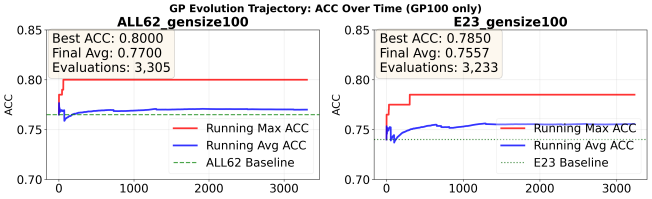

Fig. 6. GTZAN Improvement Trajectory

vergence: if fitness variance < 0.0001 over 5 generations). We evaluated the effectiveness of our proposed pipeline against established approaches, which employ XGBoost [5, 26] without GP enhancement: music mood/theme-tagging on the MTG-Jamendo dataset [27] and genre-tagging on the GTZAN dataset [28]. The mood/theme subset of the MTG-Jamendo dataset contains 18486 songs with 56 tags. The GTZAN dataset is a collection of 1,000 audio tracks, each 30 sec long, with 100 tracks for each of its 10 featured genres.

Baseline XGBoost models are trained using E23 and ALL62 feature sets, respectively, for both MTG-Jamendo mood/theme tagging and GTZAN genre tagging, following the same train/val/test splits and other setups of the state-of-the-art interpretable methods on Github [5, 26], such as identical XGBoost hyperparameters: 70 estimators with max depth 3 and learning rate 0.1 for MTG-Jamendo multiple binary classification, and max depth 2 with learning rate 0.3 for GTZAN multiclass classification.

3.2. GP-Augmented Performance Comparison

Our reproduced baselines achieve 0.727 ROC-AUC (MTG-Jamendo) and 0.765 accuracy (GTZAN) using ALL62 features, closely matching but slightly below the original reported performance [5, 26] of 0.729 ROC-AUC and 0.79 accuracy, respectively. Table 1 presents the best-performing GP-augmented feature sets for deployment on the never-seen test set with stratified bootstrap confidence intervals ($B = 2000$, 95% CI). These results demonstrate two key findings: first, GP consistently improves upon both base feature sets across datasets, with particularly substantial gains on GTZAN (4.0-5.0% accuracy improvement). This indicates that GP effectively discovers beneficial combinations, regardless of the level of information of the base feature set. Second, our method surpasses state-of-the-art interpretable approaches [5, 26]) while approaching state-of-the-art deep learning performance (0.781 ROC-AUC on MTG-Jamendo [29], 0.84 accuracy on GTZAN [30]) with only 5 GP iterations.

To validate that the performance improvements of the best-performing GP-augmented feature sets (Table 1) reflect systematic effectiveness rather than outlier behavior, Figures 3 and 4 present the complete performance distributions across all GP-augmented feature sets. The key finding is that median GP performance consistently exceeds baselines across all configurations (population size 100 or 500), with narrow inter-quartile ranges indicating stable improvement. While the median gains are small, the findings indicate that most GP-evolved features are beneficial. Deployment models (i.e., those shown in Table 1) occupy the upper tail of an overall improved distribution, rather than being rare outliers.

3.3. GP Improvement Trajectory

To assess computational cost-efficiency, Figures 5 and 6 present the anytime trajectories of GP feature construction, showing best-so-far and running-average scores versus evaluation count for GP configuration of 100 individuals in population. Across datasets, the best-so-far curves rise steeply during the first few hundred evaluations and then flatten, while the running averages increase more gradually toward a plateau. This improvement pattern remains consistent across population sizes, with GP500 showing similar trajectories (additional plots available in our GitHub repository in Section 1). This characteristic steep-then-plateau pattern reflects evolutionary search dynamics in large combinatorial spaces: early generations explore diverse feature combinations yielding rapid gains, while later refinement produces diminishing returns as the search approaches local optima. Dataset complexity determines convergence rates. Specifically, in GTZAN's 10-class genre classification, our method achieves near-optimal performance within 300 evaluations, while in MTG-Jamendo's challenging 56-tag multilabel prediction, it requires 1000 evaluations for substantial gains, with the largest improvement (E23-GP500) extending to 8000 evaluations. These improvements are both computationally efficient (5.5 sec per evaluation for MTG-Jamendo, 1.2 sec for GTZAN on RTX 3080Ti) and practically significant, surpassing established interpretable methods while maintaining modest search budgets.

4. INTERPRETABILITY ANALYSIS ON GP FEATURES

The interpretability advantage of our GP approach is demonstrated by analysing the symbolic mathematical expressions of evolved composite features and their feature-feature and feature-operator co-occurrence patterns, extending beyond traditional importance-based methods [5, 15, 24] to reveal how base features should be combined and transformed for optimal tagging performance. Note, base feature names appear in typewriter font (e.g., `Spectral-Spread`).

4.1. Best Features Expressions

We analyze composite features from the top-performing evolved expressions to understand base feature interaction mechanisms. Table 2 shows that GP discovers heterogeneous solutions ranging from linear and nonlinear combinations to complex conditional forms, revealing multiple effective pathways for feature combination rather than convergence on a single type. Several low-complexity expressions achieve top performance alongside deeper constructs, consistent with parsimony pressure [25] that compact expressions curb

Table 2. Top GP Individual Expressions

Dataset: MTG-Jamendo (Base: ALL62, Metric: AUC)	
AUC 0.730:	$\max(0, \cos(\text{glob\_glob\_sub})) - \cos(\max(0, \text{Rhythm\_Stability})) + \text{Spectral\_Energyband\_Low}$
AUC 0.729:	$\text{Loudness} - \text{BPM}^{\text{dom\_dom}}$
Dataset: GTZAN (Base: ALL62, Metric: ACC)	
ACC 0.805:	$\frac{\max(\text{sub\_sub\_dom}, \text{glob\_dom\_dom})}{\text{Danceability}} \mid (\text{glob\_dom}^{\text{Spectral\_Energyband\_Middle\_Low}})^2$
ACC 0.800:	$\text{if}\Big(\text{if}(\text{Mode} > 0, \cosh(\text{Chords\_Changes\_Rate}), \frac{\text{Spectral\_Entropy}}{\text{dom\_dom\_tonic}}) > 0, \frac{1}{1+e^{-\text{glob\_sub}}}, \frac{1}{\text{Chords\_Number\_Rate} - \text{glob\_dom\_tonic}}\Big)$
Dataset: MTG-Jamendo (Base: E23, Metric: AUC)	
AUC 0.724:	$2 \cdot \text{Length} - 3 \cdot \text{Onset\_Rate} - \min(\text{Danceability}, \text{Spectral\_Decrease})$
AUC 0.724:	$\tanh\Big(\text{Length} - 2 \cdot \text{Onset\_Rate} - \text{Zerocrossingrate}\Big)$
Dataset: GTZAN (Base: E23, Metric: ACC)	
ACC 0.790:	$\tan(\max(\text{Spectral\_Flux}, \text{Spectral\_Rolloff})) \mid \frac{1}{1+e^{-(\cosh(\text{Spectral\_Energyband\_Middle\_Low}))^2}}$
ACC 0.785:	$\text{Chords\_Number\_Rate} \cdot \text{Danceability} \mid \frac{\text{Onset\_Rate}}{\text{Dynamic\_Complexity}} - \text{Danceability} + \text{Onset\_Rate}$

bloat and are easier to interpret. For example, the MTG-Jamendo expression $\texttt{Loudness} - \texttt{BPM}^{\texttt{dom\_dom}}$ potentially captures perceived energy through dynamics-tempo interaction modulated by harmonic tension. Moreover, conditional operators could suggest piecewise or categorical feature-label correlations, such as GTZAN's conditional expression using `Mode` and `Chords_Changes_Rate` could exploit genres' characteristic differences in harmonic rhythm and major/minor tendencies.

4.2. Base Features and Operation Co-occurrence

Co-occurrence analysis reveals task-specific feature synergies with music-theoretic implications. Figure 7 shows pairwise co-occurrence (lower triangles) and mean performance conditioned on pair presence (upper triangles) for top-500 evolved expressions. For MTG-Jamendo, `Spectral-Spread` paired with timbral features appears at moderate frequency with high ROC-AUC, potentially capturing how spectral distribution affects perceived brightness and warmth. Additionally, `Spectral-Decrease` with `Beats-Loud` shows mid-range frequency but strong performance, suggesting rhythmic and spectral characteristics jointly contribute to mood discrimination. GTZAN exhibits different synergies. It benefits from harmonic function pairs (e.g., `dom-tonic-dom` with `glob-dom-glob` showing high mean accuracy), reflecting genre-distinctive chord progression patterns that GP discovers as harmonically-informed combinations. The frequent co-occurrence of `Spectral-Entropy` with `Spectral-Flux` indicates that spectral irregularity measures together enhance genre classification.

Operator-feature patterns shown in Figure 8 reveal perceptually-motivated transformations across tasks. For MTG-Jamendo, logarithmic operations on temporal features (`Danceability`, `Onset-Rate`, `Length`) appear frequently with high ROC-AUC, potentially reflecting rhythmic perception's nonlinear nature, consistent with logarithmic scaling observed in many perceptual domains. Sigmoid/swish operations with `Vocal-Instrumental` features at lower frequency but high performance suggest nonlinear vocal-instrumental balance contributes to mood perception

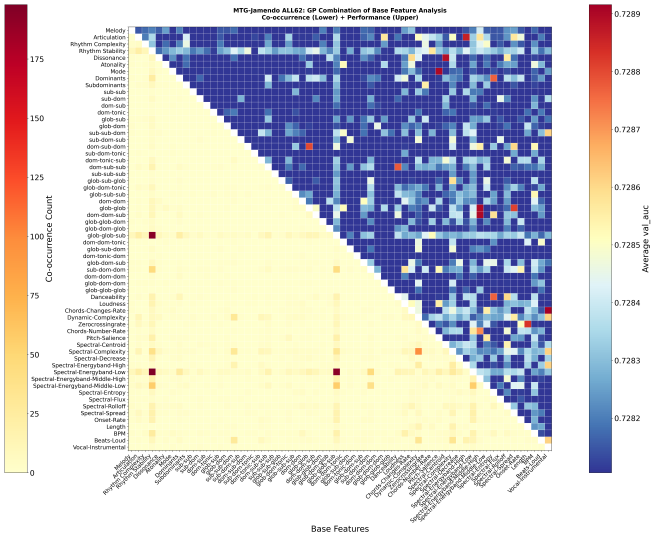

Fig. 7. MTG-Jamendo ALL62 Co-occurrence Matrix

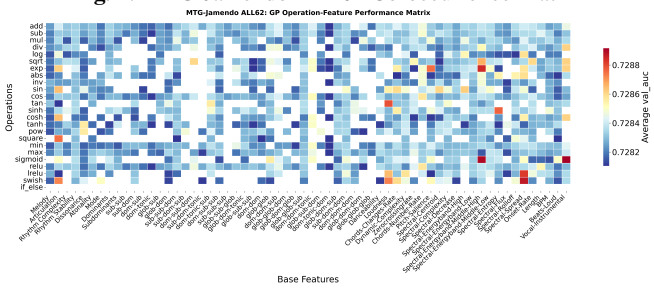

Fig. 8. MTG-Jamendo ALL62 Operator-Feature Performance

through threshold-like mechanisms. GTZAN shows different patterns. Sigmoid transforms on spectral features (`Spectral-Flux`, `Spectral-Rolloff`) achieve near-best accuracy, possibly modeling threshold effects in timbre perception, while power operations with harmony features (`glob-dom`, `glob-sub-dom`) further support the genre-harmonic relationship. (Corresponding GTZAN plots are available in our GitHub repository.)

5. CONCLUSION

This paper proposes a GP pipeline to construct interpretable composite features that augment base feature sets for music tagging. Our method mathematically combines base music features at scale to capture synergistic interactions, thereby achieving representational benefits similar to deep learning–style feature fusion. However, unlike the latter methods, our GP pipeline preserves interpretability. Experiments on MTG-Jamendo and GTZAN datasets show consistent performance gains across base features of varying abstraction levels, with improvements emerging within the first few hundred GP evaluations, and hence with a relatively modest search budget. The evolved best-performing expressions range from linear and nonlinear to conditional forms, including low-complexity solutions at top performance, showing the GP's parsimony pressure produces simpler, more interpretable expressions without sacrificing effectiveness. Analysis of feature-feature and feature-operator co-occurrence patterns reveals that interpretable feature construction extends beyond traditional importance-based methods.

6. REFERENCES

[1] Minz Won, Keunwoo Choi, and Xavier Serra, "Semi-supervised music tagging transformer," *arXiv preprint arXiv:2111.13457*, 2021.

[2] George Tzanetakis and Perry Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[3] Matthew Prockup, Andreas F Ehmann, Fabien Gouyon, Erik M Schmidt, and Youngmoo E Kim, "Modeling musical rhythmatscale with the music genome project," in *WASPAA*, 2015.

[4] Yudong Zhao, György Fazekas, and Mark Sandler, "Violinist identification using note-level timbre feature distributions," in *ICASSP*. IEEE, 2022, pp. 601–605.

[5] Vassilis Lyberatos, Spyridon Kantarelis, Edmund Dervakos, and Giorgos Stamou, "Perceptual musical features for interpretable audio tagging," in *ICASSPW*. IEEE, 2024.

[6] Jordi Pons Puig, Oriol Nieto Caballero, Matthew Prockup, Erik M. Schmidt, Andreas F. Ehmann, and Xavier Serra, "End-to-end learning for music audio tagging at scale," in *ISMIR*, 2018.

[7] Jordi Pons and Xavier Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *arXiv preprint arXiv:1909.06654*, 2019.

[8] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM TASLP*, vol. 28, pp. 2880–2894, 2020.

[9] Peter Harrison and Marcus T Pearce, "Simultaneous consonance in music perception and composition.," *Psychological Review*, vol. 127, no. 2, pp. 216, 2020.

[10] Malinda J McPherson, Sophia E Dolan, Alex Durango, Tomas Ossandon, Joaquín Valdés, Eduardo A Undurraga, Nori Jacoby, Ricardo A Godoy, and Josh H McDermott, "Perceptual fusion of musical notes by native amazonians suggests universal representations of musical intervals," *Nature communications*, vol. 11, no. 1, pp. 2786, 2020.

[11] Pei-Chun Chang, Yong-Sheng Chen, and Chang-Hsing Lee, "IIOF: Intra-and inter-feature orthogonal fusion of local and global features for music emotion recognition," *Pattern Recognition*, vol. 148, pp. 110200, 2024.

[12] Aizhen Liu, "Research on multi-feature fusion music emotion classification method under cognitive psychology," *International Journal of High Speed Electronics and Systems*, p. 2540339, 2025.

[13] Morgan Buisson, Pablo Alonso-Jiménez, and Dmitry Bogdanov, "Ambiguity modelling with label distribution learning for music classification," in *ICASSP*, 2022.

[14] Hendrik Vincent Koops, W Bas De Haas, John Ashley Burgoyne, Jeroen Bransen, Anna Kent-Muller, and Anja Volk, "Annotator subjectivity in harmony annotations of popular music," *JNMR*, vol. 48, no. 3, pp. 232–252, 2019.

[15] Saumitra Mishra, Bob L Sturm, and Simon Dixon, "Local interpretable model-agnostic explanations for music content analysis.," in *ISMIR*, 2017, vol. 53, pp. 537–543.

[16] Andre Holzapfel, Bob Sturm, and Mark Coeckelbergh, "Ethical dimensions of music information retrieval technology," *TISMIR*, vol. 1, no. 1, pp. 44–55, 2018.

[17] Bin Cui, Jialie Shen, Gao Cong, Heng Tao Shen, and Cui Yu, "Exploring composite acoustic features for efficient music similarity query," in *ACM MM*, 2006.

[18] Toni Mäkinen, Serkan Kiranyaz, Jenni Raitoharju, and Moncef Gabbouj, "An evolutionary feature synthesis approach for content-based audio retrieval," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2012, no. 1, pp. 23, 2012.

[19] Yi Mei, Qi Chen, Andrew Lensen, Bing Xue, and Mengjie Zhang, "Explainable artificial intelligence by genetic programming: A survey," *IEEE TEVC*, vol. 27, no. 3, pp. 621–641, 2022.

[20] Tingting Yang, Chenhao Xue, and Jun Chen, "Design of driver stress prediction model with CNN-LSTM: Exploration of feature space using genetic programming," in *IJCNN*. IEEE, 2024.

[21] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez Gutiérrez, Sankalp Gulati, Perfecto Herrera Boyer, Oscar Mayor, Gerard Roma Trepat, Justin Salamon, José Ricardo Zapata González, and Xavier Serra, "Essentia: An audio analysis library for music information retrieval," 2013.

[22] Yu-Te Wu, Yin-Jyun Luo, Tsung-Ping Chen, I Wei, Jui-Yang Hsu, Yi-Chin Chuang, Li Su, et al., "Omnizart: A general toolbox for automatic music transcription," *arXiv preprint arXiv:2106.00497*, 2021.

[23] Andy Seaborne and Eric Prud'hommeaux, "SPARQL query language for RDF," *W3C Recommendation, W3C*, 2008.

[24] Tianqi Chen and Carlos Guestrin, "XGBoost: A scalable tree boosting system," in *KDD*, 2016, pp. 785–794.

[25] Riccardo Poli and Nicholas Freitag McPhee, "Parsimony pressure made easy," in *GECCO*, 2008.

[26] Vassilis Lyberatos, Spyridon Kantarelis, Edmund Dervakos, and Giorgos Stamou, "Challenges and perspectives in interpretable music auto-tagging using perceptual features," *IEEE Access*, 2025.

[27] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra, "The MTG-Jamendo dataset for automatic music tagging," ICML, 2019.

[28] Bob L Sturm, "The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use," *arXiv preprint arXiv:1306.1461*, 2013.

[29] Jaeyong Kang and Dorien Herremans, "Towards unified music emotion recognition across dimensional and categorical models," *arXiv preprint arXiv:2502.03979*, 2025.

[30] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino, "Masked modeling duo: Learning representations by encouraging both networks to model the input," in *ICASSP*, 2023, pp. 1–5.

[31] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner Gardner, Marc Parizeau, and Christian Gagné, "Deap: Evolutionary algorithms made easy," *JMLR*, vol. 13, no. 1, pp. 2171–2175, 2012.