A Kolmogorov Arnold Network NAS Framework for Strumming Pattern Recognition in Technology-Enhanced Pop/Rock Music Education

Marco Pennese
Conservatory of Music "G. Puccini"
La Spezia, Italy
marco.pennese@steffani.it

Stefano Giacomelli
DISIM, University of L'Aquila, Italy
L'Aquila, Italy
stefano.giacomelli@graduate.univaq.it

Claudia Rinaldi
DISIM, University of L'Aquila / CNIT
L'Aquila, Italy
claudia.rinaldi@univaq.it

***Abstract*—This paper contributes to the Internet of Sounds by presenting a modular neural architecture for automatic classification of guitar strumming patterns in pop/rock music, developed within the *Strummin' Gym* project, an interactive platform for technology-enhanced music education. Motivated by the observation that a small set of rhythmic templates recurs across the genre, we frame a novel multinomial classification task to support strumming analysis and instrument learning. We construct and release a curated dataset of annotated audio excerpts from YouTube and extract diverse rhythmic descriptors using LibROSA, Essentia, and the TU-Wien Rhythmic Pattern Extractor. The proposed Neural Networks model combines 1D/2D convolutional-attentive encoders with Kolmogorov–Arnold Networks, optimized via Neural Architecture Search and Hyperparameter Tuning. Results show that KANs effectively capture structured audio regularities and confirm the emergence of recurring strumming practices. This work exemplifies how machine listening can enhance IoS applications for inclusive and scalable music education.**

***Index Terms*—Internet of Sounds (IoS), Music Information Retrieval (MIR), Strumming Patterns Classification, Deep Learning (DL), Convolutional Neural Networks (CNNs), Self-Attention (SA), Kolmogorov–Arnold Networks (KANs), Neural Architecture Search (NAS), Hyperparameter Optimization (HPO), Digital Signal Processing (DSP)**

I. Introduction

The convergence of sound technologies, intelligent systems, and networked infrastructures has led to the emergence of the *Internet of Sounds* (IoS) paradigm [1]. This interdisciplinary framework envisions a distributed ecosystem of connected audio devices — *Sound Things* — capable of capturing, processing, transmitting, or generating sound in *real time*. IoS applications span acoustic monitoring [2], auditory surveillance [3], interactive musical systems, and augmented audio experiences [4]. Within this landscape, the *Internet of Musical Things* (IoMusT) has gained traction, emphasizing smart musical objects and Networked Music Performance (NMP) for enhanced human–machine interaction in artistic and educational contexts [5].

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

This paper operates at the intersection of IoMusT and *machine listening* for music understanding. We introduce a *neural* framework for automatic classification of guitar strumming patterns in pop/rock music, designed as a core module of an intelligent rhythm-feedback system within a broader IoS infrastructure (Figure 1). Detecting rhythmic accompaniment patternsenables interactive support for self-directed guitar learners and technology-enhanced pedagogy. Our architecture integrates lightweight convolutional-attentive encoders with Wavelet-based Kolmogorov–Arnold Networks (WavKANs), prioritizing modularity and efficiency for embedded or edge-level IoS deployments.

(1) We propose a scalable framework for Neural Architecture Search (NAS) and Hyperparameter Optimization (HPO) tailored to rhythm-based audio classification [1].

(2) We provide one of the first empirical validations of KANs for structured audio modeling, supporting efficient and interpretable learning.

(3) We publicly release a curated dataset of pop/rock excerpts annotated with categorical strumming patterns [6], fostering reproducibility and benchmarking within Music Information Retrieval (MIR) and IoS research.

By embedding this system into the IoS vision, we aim to promote intelligent, network-aware musical tools for inclusive, personalized, and scalable music education.

II. The Strummin' Gym Project

Learning to play guitar — whether under the guidance of a teacher, through peer interaction, or via self-directed study — requires substantial individual practice to achieve musical and *groove* fluency. While instructors, method books, tutorials, and online resources offer valuable support, a persistent gap remains in the monitoring and assessment of performance during self-guided practice. This underscores the need for technological solutions that can assist and guide *novice guitarists* (and their instructors) during the early stages of musical development. Focusing on the acquisition of strumming techniques in the pop/rock genre, the *Strummin' Gym* project proposes

[1] https://github.com/StefanoGiacomelli/StrumKANet

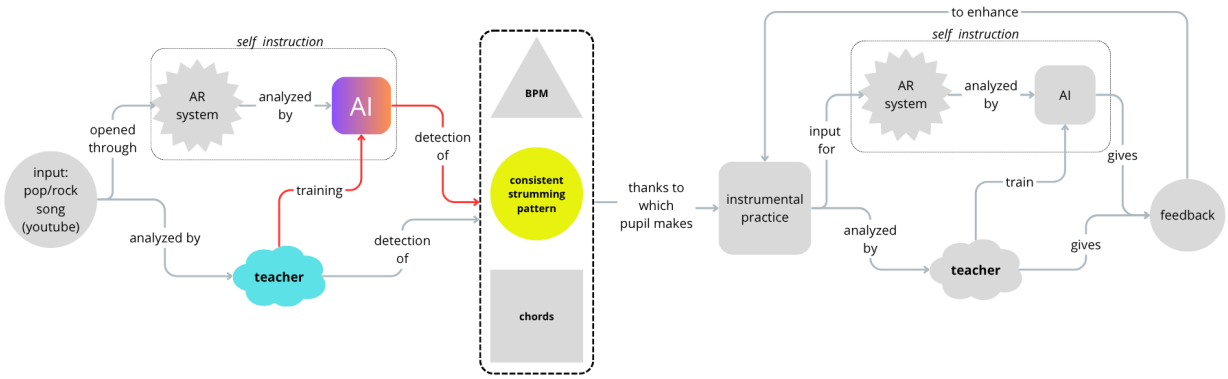

Fig. 1. The Strummin' Gym project framework: Phase 2 highlight.

a web-based platform that structures the learning process around a small set of operational parameters (Figure 1). Once internalized, these parameters enable beginners to deviate from predefined models and progressively develop their own expressive style.

Phase 1. The system provides structured training aimed at mastering the core components of rhythmic accompaniment: tempo (BPM), strumming pattern, chord progression, and pattern repetition. A sequence of increasingly complex exercises is delivered either by the system or an instructor, while a Neural Network (NN) module offers real time feedback to support user performance.

Phase 2. The user submits a pop/rock audio track of their choice. A dedicated NNs module automatically extracts its BPM, a consistent strumming pattern, chord sequence, and the corresponding repetition structure. As in Phase 1, a second network delivers feedback to enhance execution accuracy. The system is designed for accessibility across both low-immersion devices and augmented reality (AR) interfaces, enabling flexible and adaptive interaction.

This paper focuses on *Phase 2*, specifically on the design, training, and evaluation of the NN model responsible for extracting a consistent strumming pattern from the user-selected audio track. Unlike traditional tasks such as tempo estimation [7], [8], beat and downbeat tracking [9]–[11], or onset detection [12], our approach addresses a novel and underexplored task: the association of a *categorical strumming pattern* with an input audio excerpt. Notably, the strumming patterns considered in this work may either be explicitly performed by a rhythm guitar or perceptually induced by the track's underlying *Tactus*, even in the absence of actual strumming gestures, reflecting stylistic nuances typical of the pop/rock music genre. In such contexts, the fretting hand forms chords while the strumming hand articulates rhythmic motion through alternating downstrokes and upstrokes. Depending on tempo, meter, and beat subdivision, a broad variety of strumming patterns has emerged across styles and artists. However, from a pedagogical perspective, a reduced set of core patterns is widely adopted for instructional purposes. According to [13], three fundamental patterns — two in binary meter and one in ternary — are sufficient to represent most strumming behaviors encountered in pop/rock (Figure 2). The proposed classification task thus targets these mid-level rhythmic structures, which are specific to accompaniment practices and offer a complementary perspective not captured by downstream MIR tasks.

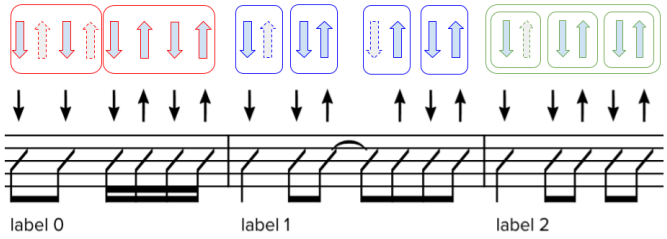

Fig. 2. Pulse subdivision and corresponding strumming patterns.

Despite the extensive availability of datasets for beat, tempo, onset, and even expressive drumming analysis, to the best of our knowledge no public dataset currently exists that provides reliable annotations of categorical strumming patterns aligned with audio excerpts (Table I). This lack of dedicated resources further highlights the novelty and pedagogical relevance of the task addressed in this work.

A. The Strummin' Dataset: a Pop/Rock Curated Selection

To support research on rhythmic pattern classification in pop/rock music, we introduce the *Strummin' Dataset* [6], a curated resource designed to meet both our pedagogical and computational requirements. The dataset originates from the first author's extensive experience in entry-level music education. A representative selection of international and italian pop/rock songs was compiled to reflect stylistic diversity and rhythmic salience. Each song entry in the dataset is annotated in a `.csv` file containing: **Metadata:** title, performer, and official YouTube audio URL; **BPM Annotations:** three tempo references — (I) `bpm_marco`, manually annotated; (II) `bpm_tunebat`, via Tunebat [2], using `Essentia.js` [20],

[2]https://tunebat.com/Analyzer

TABLE I
COMPARATIVE OVERVIEW OF RHYTHM ANALYSIS DATASETS (ISMIR, SCOPUS, IEEE, GOOGLE SCHOLAR)

Dataset	Year	Tasks	Audio Samples	Annotations	License
Carnatic Music Rhythm [14]	2014	Meter tracking	CMD: 118 recs × 2min; CMDf: 176 tracks (∼16.6h)	Human beat/downbeat, 4 cycles	By request (Zenodo)
Candombe Recordings [15]	2015	Beat/downbeat tracking	35 tracks (∼2h, 44.1kHz)	4700 human downbeat	CC-BY 4.0 (Official site)
Extended Ballroom [16]	2016	Dance rhythms recognition	3992 clips × 30s, 9 classes	Manual annotations of rhythm, genre	Open (IRCAM page)
GiantSteps–Tempo [8], [17]	2015/18	Tempo estimation, Ambiguity analysis	664 clips tempo + 604 clips key (60–120s, MP3)	Crowdsourced annotation, experts validation; (time, key)	CC-BY 4.0 (GitHub)
Groove MIDI Dataset [18]	2019	Grooves, infilling, expressivity modeling	1150 MIDI files (∼13.6h), synthetic aligned audio	Hits, offset, velocity, performer, style	CC-BY 4.0 (Magenta page)
GuitarSet [10]	2018	Automatic transcription, beat/downbeat, stroke detection	180 excerpts (∼3h), 7chs (pickup + mic)	Notes, chords, tempo, beat, strokes; JAMS, JSON	CC-BY 4.0 (Official site)
The Harmonix Set [11]	2019	Beat/downbeat, structure segmentation	912 tracks (annotations)	Beat/downbeat, segments; JAMS, CSV, TXT	CC-BY 4.0 (GitHub)
MIREX2006 [7]	2006	Tempo estimation	160 clips × 30s (WAV) (∼80min)	2 perceived tempos, saliency	Open (MIREX page)
MODAL [12]	2011	Onset detection, ODF benchmark	501 onset events (44.1kHz)	Human onsets (±50ms)	GPL/CC (GitHub)
The Beatles [9]	2010	Chord recognition, beat/downbeat, key, structure segmentation	180 pieces (∼8h); annotations only (no audio)	Chord, key, beat, downbeat, segments; .lab file	CC BY-NC-SA 2.0 UK (Official site)
Million Song Dataset [19]	2011	Tagging, recognition, analysis (multiple)	1M songs; features; audio preview 30s via API	Segments, tempo, timbre/pitch vectors, tagging; HDF5, CSV	Open (Official site)

[21]; and (III) `bpm_songbpm`, via SongBPM [3] (Spotify API); **Rhythmic Label:** a human-validated category (0 = binary, 1 = quaternary, 2 = ternary — Figure 2); **Download Flag:** a binary flag indicating successful retrieval via a custom Python tool [22].

To streamline code experiments, the dataset is integrated into a `PyTorch`-based framework [23], with audio loaded via `TorchAudio` [24], [25] and segmented into 10-second windows (excluding beginning and end). Each segment: may be used as waveform or as rhythmic descriptors (Section III-A); inherits the label from the parent track; and supports both batch and *on-the-fly* feature extraction.

The dataset class is implemented as a `LightningDataModule` [26], allowing for reproducible training/validation/test splits and distributed loading (Section V). To promote open research, we release:

- the original `.csv` and all available audio files;
- a pre-processed version (`.npz`) with extracted features in Python dictionaries;
- high-resolution `.svg` plots of each feature per track.

All materials are publicly available on Zenodo [6], including metadata and usage instructions.

III. RHYTHM FEATURES AND NEURAL NETWORKS ARCHITECTURES

The automatic classification of rhythmic accompaniment patterns from audio requires informative signal representations and suitable learning architectures. We first present the adopted feature extraction strategies, focusing on *state-of-the-art* rhythm descriptors capturing temporal dynamics at multiple levels. Then, we review learning paradigms relevant to our task, discussing the motivations behind our architectural choices.

A. Rhythm Features Extraction

Effective classification of strumming patterns requires representations capturing both fine-grained temporal dynamics and mid-level periodicity. We selected a curated subset (Table III) of descriptors from three widely adopted toolkits — `LibROSA` [27], `Essentia` [28], and TU-Wien's `RP_extract` [29] — based on their complementarity and documentation quality.

[3]https://songbpm.com

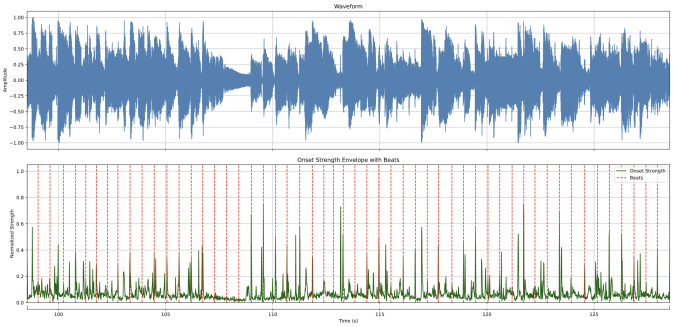

Fig. 3. Onset Strength envelope and Beats tracking for "Adele - Milion Years Ago".

Onset Strength (OS): computed from a log-power Mel spectrogram $S(f,t)$ using `n_fft`=2048 and `hop_length`=512, OS captures broadband transients by comparing each frame with a local frequency maximum [30]:

$$S^{\max}(f,t) = \max\left(S(f-1,t),\ S(f,t),\ S(f+1,t)\right)$$

$$OS(t) = \sum_{f=1}^{F} \max\left\{0,\ S(f,t) - S^{\max}(f,t-\mu)\right\}$$

where F is the number of Mel bands and μ is a fixed lag. This suppresses vibrato/legato smearing and highlights rhythmic activations (Figure 3).

Tempogram & Tempogram Ratio: the tempogram $\mathcal{T}(t,\tau)$ is computed via short-time autocorrelation of OS [31]:

$$\mathcal{T}(t,\tau) = \sum_{w=-W/2}^{W/2} OS(t+w)\, OS(t+w-\tau)$$

with τ covering 300 BPM bins. The Tempogram Ratio projects periodicities onto metrical subdivisions of the estimated tempo [32], yielding a 13-dimensional rhythmic profile (Figure 4, Table II).

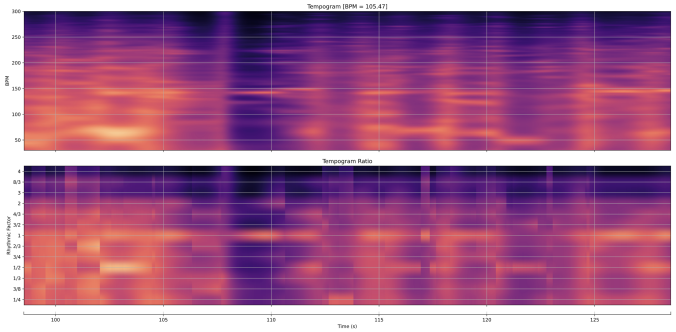

Fig. 4. Tempogram and *Spectral Rhythm Patterns* (Tempogram Ratio) for "Adele - Milion Years Ago".

TABLE II
METRICAL SUBDIVISIONS FOR TEMPOGRAM RATIO

Row index	Factor	Musical Subdivision
0	4	Sixteenth
1	8/3	Dotted sixteenth
2	3	Eighth triplet
3	2	Eighth note
4	4/3	Dotted eighth
5	3/2	Quarter triplet
6	1	Quarter note
7	2/3	Dotted quarter
8	3/4	Half triplet
9	1/2	Half note
10	1/3	Dotted half note
11	3/8	Whole triplet
12	1/4	Whole note

Beat Tracking: based on dynamic programming [33], beats are detected by maximizing the cumulative score

$$S(B) := \sum_{\ell=1}^{L} \Delta(b_\ell) + \lambda \sum_{\ell=2}^{L} P_{\hat{\delta}}(b_\ell - b_{\ell-1})$$

where B is a beat sequence, $\Delta(\cdot)$ is the novelty function from OS, $\hat{\delta}$ the estimated beat period, and $P_{\hat{\delta}}$ a tempo deviation penalty. The beat sequence anchors high-level rhythm descriptors, as visualized in Figure 5.

Developed by TU Wien, the `RP_extract` library yields psychoacoustically informed rhythm features (Figure 6):

- **Rhythm Pattern (RP)**: a 1440-dimensional matrix describing energy modulations across Bark bands and modulation frequencies. Computed via auditory preprocessing (Bark $\rightarrow$ Phon $\rightarrow$ Sone) and modulation DFT (Figure 6).
- **Rhythm Histogram (RH)**: global summary of rhythmic energy per modulation bin (60D), obtained by collapsing RP over Bark bands.
- **Temporal Statistical Spectrum Descriptor (TSSD)**: 1176D vector capturing temporal variation in spectrum statistics (*mean, median, variance, skewness, kurtosis, minimum, maximum*) over time, per Bark band.
- **Temporal Rhythm Histogram (TRH)**: 420D vector capturing the same statistical descriptors over randomly sampled RH segments.

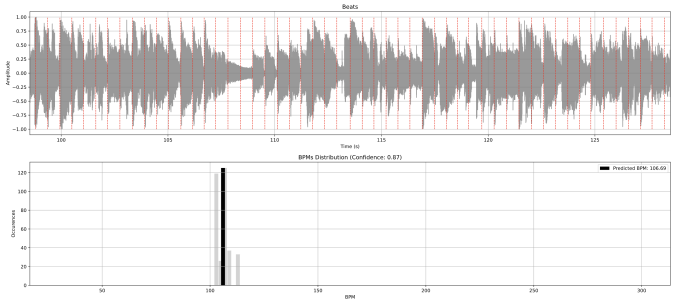

Fig. 5. Beats and BPMs distribution for "Adele - Milion Years Ago".

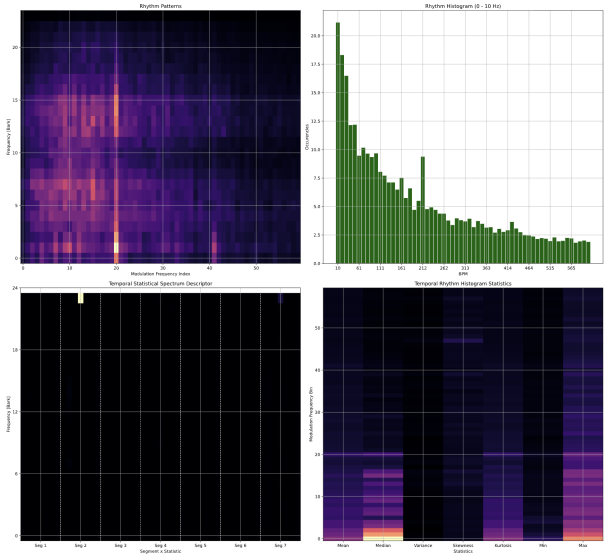

Fig. 6. Rhythm Pattern (top-left), Rhythm Histogram (top-right), TSSD (bottom-left), and TRH (bottom-right) for "Adele - Milion Years Ago".

B. Kolmogorov-Arnold Networks for Audio Modeling

KANs have recently emerged as flexible alternatives to standard feed-forward layers, replacing fixed activation functions with learnable B-spline operators. While effective in symbolic domains, their performance on audio tasks remains limited. A benchmark by Yu et al. [34] showed KANs underperforming MLPs on datasets like UrbanSound8K and Speech-Commands, highlighting challenges in capturing perceptual and hierarchical audio patterns via spline-based activations. To address these limitations, Zhang et al. [35] proposed the Kolmogorov–Arnold Fourier Network (KAF), replacing B-splines with trainable Random Fourier Features and hybrid Gaussian Error Linear Units (GELU)–Fourier activations. KAF improved spectral sensitivity and parameter efficiency, outperforming both KANs and MLPs on various audio benchmarks. Its hybrid nonlinearity modulates frequency focus across layers, enhancing acoustic modeling. Building on this, Koudounas et al. [36] integrated KANs into CNN/Transformer architectures for Spoken Language Understanding, testing five injection strategies. The FKF variant (KAN between feed-

TABLE III
SUMMARY OF RHYTHM FEATURES PRE-PROCESSING

Features	Out Tensor Shapes	Dim.	Pre-Processing Parameters
Raw input waveform	(B, T)	1D	**sample_rate**: 48kHz
Onset Strength (OS)	(B, T)	1D	`LibROSA`
Tempogram	$(B, 384, T)$	2D	**lag**: 1, **fft_size**: 2048, **hop_size**: 512,
Tempogram Ratio	$(B, 13, T)$	2D	**win_type**: hann, **n_filts**: 128, **norm**: area,
Beats	(B, N_{beats})	1D	**interp**: linear
Rhythm Pattern (RP)	$(B, 24, 60)$	2D	`RP_extract`
Rhythm Histogram (RH)	$(B, 60)$	1D	**win_type**: hann, **win_size**: 1024, **hop_size**: 512
Temporal Statistical Spectrum Descriptor (TSSD)	$(B, 24, 49)$	2D	**bark_bands**: 24, **mod_amp**: 60
Temporal Rhythm Histogram (TRH)	$(B, 60, 7)$	2D	**transforms**: to Bark→to dB→to Phon→to Sone
Beats	(B, N_{beats})	1D	same as `LibROSA`
BPM Distribution Estimates	(B, N)	1D	

forward layers) proved most effective across multilingual corpora. Their analysis also confirmed that B-splines offered the best balance between interpretability and accuracy. KAN-augmented Transformers achieved improved attention alignment to semantically salient audio regions, enhancing both performance and explainability. Phuong et al. [37] introduced Group-Rational KANs (GR-KAN) within XLSR-Conformer backbones for Synthetic Speech Detection, replacing MLPs with GR-KANs to reduce Equal Error Rate by up to 67.1% on ASVspoof2021. GR-KAN's variance – preserving initialization and dimensionality reduction – supported improved generalization across SSL embeddings. KANs have also shown promise in multimodal fusion: Talha et al. [38] employed KANs to fuse MFCC and emotion-related embeddings (from AffectNet), achieving 97.67% on CREMA-D and improving interpretability over MLPs. Similarly, Zheng et al. [39] introduced ICKAN, a CNN–KAN hybrid for musical instrument recognition, reaching 95.74% accuracy via expressive timbral modeling.

These works suggest that while standalone KANs may struggle in audio domains, they become effective when paired with suitable inductive biases (e.g.: convolution, attention) or used in fusion/projective roles. We build on this insight by proposing a lightweight CNN+Att+KAN hybrid optimized for rhythm classification. Leveraging NAS and HPO, we demonstrate that compact, wavelet-based KANs can achieve competitive performance with reduced complexity, supporting KAN adoption for descriptive audio modeling.

IV. IMPLEMENTATION DETAILS

This section describes the architectural components of our framework for automatic strumming pattern classification. We detail the convolutional encoders used for rhythmic feature processing and the Wavelet KAN (WavKAN) classifier, all organized in a modular pipeline optimized via Neural Architecture Search (NAS) and Hyper-parameters Optimization (HPO, Sections IV-C – V-A).

A. Convolutional Encoders

We designed lightweight 1D and 2D convolutional encoders to project rhythmic features into compact embeddings. Each module optionally integrates attention mechanisms to capture temporal and spatial structure.

1) 1D Encoders: applied to sequential features such as OS, Beats, BPMs, and raw waveforms, include batch-normalized 1D convolutions, Rectified Linear Unit (ReLU), adaptive pooling, and optionally the following. **Sinusoidal Positional Embedding (SPE)** [40]: deterministic encoding that augments the temporal dimension with sine and cosine functions, enabling relative and absolute position awareness. Given $x \in \mathbb{R}^{B\times C\times T}$:

$$SPE_{c,t} = \begin{cases} \sin\left(\frac{t}{10000^{\frac{c}{C}}}\right), & c \text{ even} \\ \cos\left(\frac{t}{10000^{\frac{c-1}{C}}}\right), & c \text{ odd} \end{cases} \qquad y = x + SPE$$

Squeeze-and-Excitation (SE) [41]: a channel-wise attention mechanism using global average pooling (GAP), followed by a bottleneck MLP and a sigmoid gating function. The resulting weights modulate the input tensor:

$y = x \cdot \sigma\left(W_2 \cdot \mathrm{ReLU}(W_1 \cdot \mathrm{GAP}(x))\right)$ where W_1 and W_2 are learnable weights.

Depthwise Convolutional Attention (CBAM) [42]: this efficient attention mechanism models local and temporal dependencies via *depthwise separable* convolutions:

$$y_c(t) = (x_c * \phi)(t) = \sum_{i=-k'}^{k'} x_c(t-i)\,\phi(i), \quad \text{for } c = 1, \ldots, C$$

with a shared kernel $\phi \in \mathbb{R}^k$ and $k' = \lfloor k/2 \rfloor$. The output is then refined via a *pointwise* convolution:

$$z(t) = \sum_{c=1}^{C} w_c\, y_c(t) + b, \quad \text{with } w_c \in \mathbb{R}$$

2) 2D Encoders: designed for 2D features like Tempogram, RP, RH, TRH, TSSD. Each encoder includes a convolution block with optional attention as follows. **Multi-Head Self-Attention (MHSA)** [43]: standard transformer-style attention:

$$\mathrm{MHSA}(x) = \mathrm{Concat}\left(\mathrm{head}_1, \ldots, \mathrm{head}_h\right) W^O$$

$$\mathrm{head}_i = \mathrm{Attn}(xW_i^Q, xW_i^K, xW_i^V)$$

with learned projection matrices $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{C\times d}$, and $W^O \in \mathbb{R}^{hd\times C}$. To manage memory with high-resolution inputs, we apply *adaptive* average pooling prior to attention,

standardizing spatial dimensions to $(H', W') = (32, 32)$. For input $x \in \mathbb{R}^{B\times C\times H\times W}$:

$$y_{b,c,i,j} = \frac{1}{|\Omega_{i,j}|} \sum_{(u,v)\in\Omega_{i,j}} x_{b,c,u,v}$$

where $\Omega_{i,j} \subseteq [1, H] \times [1, W]$ denotes the pooling window.

Axial Attention (AA) [44]: self-attention along each spatial axis, reducing MHSA quadratic complexity:

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

All encoders output flattened embeddings to feed the downstream KAN classifier. Tables III and IV maps feature types to encoder configurations.

B. Wavelet Kolmogorov–Arnold Networks

To classify strumming patterns, we adopt our `PatternKAN`, a shallow WavKAN grounded in the Kolmogorov–Arnold representation theorem and extended with wavelet-based activations [45]. WavKAN generalizes spline-based KANs [46] by replacing their piecewise-polynomial activations with parametrized wavelet functions.

Each unit in WavKAN performs a linear projection of the input followed by a nonlinear activation modeled as a linear combination of scaled and translated wavelets:

$$g(z) = \sum_{k=1}^{K} \alpha_k \psi\left(\frac{z-\mu_k}{s_k}\right), \quad \psi(z) = (1-z^2)e^{-\frac{z^2}{2}}$$

where α_k, μ_k, and s_k are learnable coefficients, shifts, and scales. This formulation enables compact, smooth, and localized function approximation, particularly suited for oscillatory and non-stationary rhythmic patterns. The architecture consists of a wavelet-activated hidden layer and a fully connected output layer. Trained end-to-end, WavKAN provides enhanced expressivity and differentiability over spline-based KANs, enabling efficient modeling of subtle rhythmic structures in musical signals.

C. Hyperparameters and Neural Architecture Search

We adopt a joint NAS–HPO strategy to explore both architectural and training-level configurations. The search space covers encoder–attention pairings for 1D/2D inputs, convolutional parameters (depth, width, kernel size), input features, and loss functions. The classifier is fixed to the `PatternKAN` architecture, resulting in 468 distinct configurations via scripting-generated `.yaml` files, ensuring full reproducibility. Details are in Table IV.

To handle ambiguity in rhythmic labeling, especially between binary patterns (e.g.: eighth- vs. sixteenth-note strumming), we introduce a *class-aware* smoothing variant of the standard Categorical Cross Entropy (CCE) loss. For class i, a softened target $\tilde{y}$ is defined using a tolerance $\tau \in [0, 1]$:

$$\tilde{y}_i(j) = \begin{cases} 1-\tau, & j = i \in \{0, 1\} \\ \tau, & i \in \{0, 1\}, j \neq i \in \{0, 1\} \\ 1, & i = j = 2 \end{cases}$$

TABLE IV
HPO - NAS SEARCH SPACE

Category	Experimental Space
1D Features	OS, Beats, BPMs distribution, RH
2D Features	Tempogram, Tempogram Ratio, RP, TSSD, TRH
1D Encoder	Channels: [16, 32]; Kernels: [3, 5, 7] Attention: None, SPE, SE, CBAM
2D Encoder	Channels: [16, 32]; Kernels: [(3,3), (5,5), (7,7)] Attention: None, AA, MHSA
Attention Params	SE: red=16; CBAM: k=3; AA/MHSA: heads=8, dim=32
Loss	Categorical Cross-Entropy with: tolerance = 0.3; smoothing = [True, False]

The smoothed loss is:

$$\mathcal{L}_{\text{smooth}} = -\sum_{j=1}^{C} \tilde{y}(j) \log p_j$$

This formulation should reduce overconfidence and better reflect human perception of close rhythmic variants. We assess its impact on classification robustness in Section V.

V. RESULTS DISCUSSION

To ensure reproducibility, scalability, and efficient orchestration of our experimental pipeline, we developed an automated framework based on structured `.yaml` configurations, modular pre-processing, and streamlined training/evaluation routines using `PyTorch Lightning` [26]. Each experiment is uniquely defined by its configuration file and managed via a central `.csv` log, which tracks completed runs and prevents duplication. For every pending configuration, the following runtime pipeline is executed:

Preprocessing & Feature Extraction: descriptors are computed on-the-fly using `LibROSA`, `RP_extract`, or `Essentia`, depending on the selected features (Section III-A). When pre-processing is disabled, the raw monophonic waveform is used directly.

Dataset Setup: audio excerpts are sampled from our custom dataset (Section II-A), pre-segmented into 10s clips at 48kHz. Dataset splits follow an 80% - 10% - 10% train/val/test partition. Mini-batches of $B = 8$ samples are shuffled and loaded using 4 parallel workers. Random seeds are fixed to 42 for full reproducibility across CPU and GPU runs.

Model Initialization: a dummy batch is propagated through the selected encoder (1D or 2D) to infer the output dimensionality and initialize the downstream WavKAN classifier (Section IV-B).

Training & Evaluation: models are trained using Adam [47] optimizer with fixed hyperparameters ($\eta = 5\times10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-4}). To enable fast benchmarking across the 468 NAS-HPO configurations (Section IV-C), each model is trained for a single epoch — a constrained regime designed to reveal early generalization trends under limited training exposure. Logs are managed via TensorBoard [4].

Results Aggregation: test loss and accuracy for each run are appended to the global log. Each experiment is consistently

[4]https://www.tensorflow.org/tensorboard

indexed by its `experiment_id` and excluded from future runs once completed.

This design supports efficient exploration of architectural trends under minimal training time, enabling early insights into which configurations offer the best trade-off between inductive bias and optimization stability. However, we acknowledge an inherent limitation: the most successful models in this setting may reflect faster convergence rather than superior long-term generalization. As no established benchmarks yet exist for training KANs on audio tasks, this protocol should be interpreted as a first exploratory step toward understanding their training dynamics in music rhythm classification contexts.

A. NAS-HPO Ablation study

TABLE V
NAS-HPO TOP-10 MODELS BY TEST ACCURACY

Encoder Params	Attention	Feature	Smoothing	Accuracy
chs:16 ks:[7, 7]	**None**	**RP**	**True**	**0.6898**
chs:16 ks:[5, 5]	None	RP	False	0.6417
chs:32 ks:[7, 7]	None	RP	False	0.6417
chs:16 ks:[5, 5]	None	RP	True	0.6328
chs:32 ks:[7, 7]	None	RP	True	0.6310
chs:16 ks:7	CBAM	OS	False	0.6168
chs:16 ks:5	CBAM	RH	True	0.6132
chs:32 ks:[7, 7]	None	T. Ratio	False	0.6114
chs:16 ks:[3, 3]	None	RP	False	0.6096
chs:16 ks:5	None	OS	False	0.6096

Grey – Common configurations for *Accuracy* and *Loss* ordering.

TABLE VI
NAS-HPO TOP-10 MODELS BY TEST LOSS

Encoder Params	Attention	Feature	Smoothing	Loss
chs:16 ks:[5, 5]	**None**	**RP**	**False**	**0.7805**
chs:32 ks:[7, 7]	None	RP	False	0.7845
chs:16 ks:[7, 7]	None	RP	True	0.8213
chs:32 ks:[7, 7]	MHSA	T. Ratio	False	0.8508
chs:32 ks:[3, 3]	None	RP	False	0.8563
chs:32 ks:[7, 7]	None	T. Ratio	False	0.8566
chs:16 ks:[5, 5]	None	RP	True	0.8654
chs:16 ks:[5, 5]	None	T. Ratio	False	0.8694
chs:32 ks:[5, 5]	None	RP	False	0.8722
chs:16 ks:[3, 3]	None	RP	False	0.8741

Grey – Common configurations for *Accuracy* and *Loss* ordering.

Figure 7 summarizes the performance landscape of the entire NAS-HPO campaign by plotting test accuracy against loss. As expected, high-performing models cluster in the top-left quadrant, confirming an inverse correlation between the two metrics. Attention mechanisms show distinct behaviors (Figure 8): models using no attention or SPE generally achieve the most favorable trade-off between accuracy and stability. CBAM yields competitive accuracy in specific configurations, especially when paired with features rich in local saliency, while MHSA shows lower median accuracy and higher variance, indicating limited suitability under current constraints.

Tables V and VI list the top-10 configurations ranked by accuracy and loss, respectively. Simpler encoder variants — featuring reduced channel depth and kernel size — consistently outperform attention-augmented counterparts. This trend holds across both evaluation metrics, suggesting that

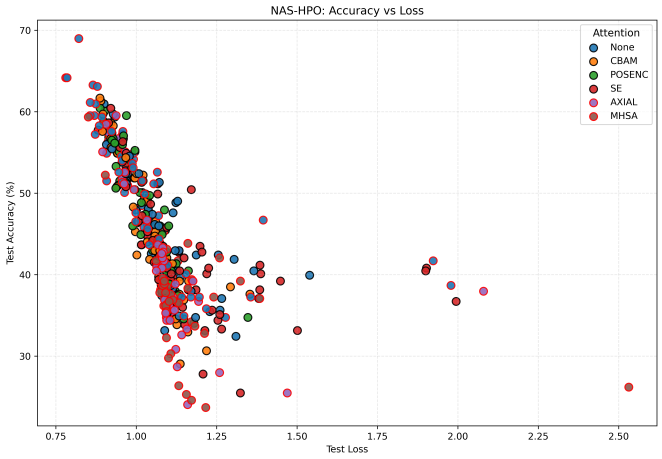

Fig. 7. Test Accuracy VS Test Loss for our NAS-HPO experiment. Note: `POSENC` stands for *Sinusoidal Positional Encoding* (SPE).

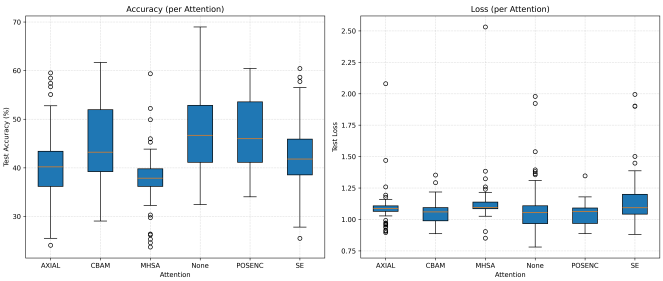

Fig. 8. Encoders Attention-related Box-Plots. Note: `POSENC` stands for *Sinusoidal Positional Encoding* (SPE).

lightweight convolutional modules may already provide sufficient inductive bias for the rhythmic pattern classification task.

We further analyze performance variability across input features in Figure 9. RP, OS, and RH exhibit higher median accuracy and tighter variance, confirming their discriminative power for rhythmic structure modeling. Conversely, descriptors such as TRH, TSSD, and Beats yield more dispersed results, likely due to their lower temporal resolution or more abstract representations. Interestingly, raw waveform models achieve competitive accuracy in isolated cases, albeit with

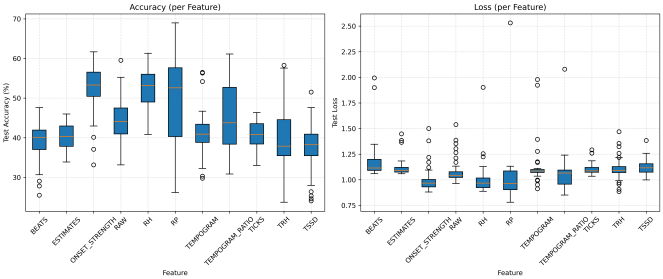

Fig. 9. Input Feature effectiveness Box-Plots.

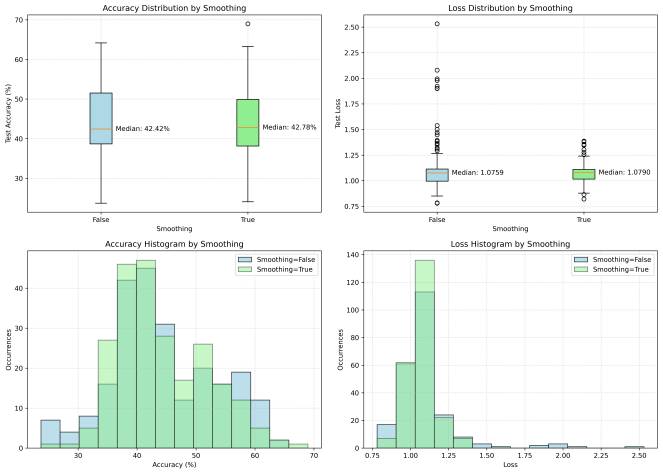

Fig. 10. Class-aware Smoothing mechanism effectiveness analysis.

higher outcome variability.

The effect of our class-aware smoothing mechanism is detailed in Figure 10. Despite negligible differences in median test accuracy (42.42% without smoothing vs. 42.78% with) and loss (1.0759 vs. 1.0790), smoothed models exhibit reduced interquartile ranges, indicating more stable predictions. Histogram analysis reveals that smoothing concentrates accuracy scores within the 40–50% band and reduces outlier frequency. Test loss distributions are similarly peaked around 1.0–1.1, with smoothing yielding slightly sharper concentration. These results suggest that smoothing helps mitigate the effect of borderline rhythmic classes without introducing systematic bias — a behavior consistent with its design goal of encouraging tolerance to perceptual ambiguities rather than boosting overall accuracy.

Based on this exploratory analysis — conducted under a constrained single-epoch regime — we selected the top-performing models by merging the top entries from Tables V and VI, yielding 14 candidate configurations. This subset, referred to as `PatternKAN`, constitutes the experimental foundation for the subsequent training phase, enabling a more focused assessment of generalization capabilities in KAN-based audio classifiers.

B. Baseline Models Training

Compared to the earlier NAS-HPO stage, the `PatternKAN` training protocol adopts a more structured and exhaustive approach, focusing on long-term model optimization rather than preliminary benchmarking. Each selected configuration is trained for up to 50 epochs, with early stopping enabled (patience = 20 epochs without validation accuracy improvement), allowing models to reach effective convergence. Checkpointing mechanisms retain the best-performing models, which also serve as dataset-related baselines [5]. Dataset splits, preprocessing, and feature extraction procedures remain consistent with the NAS-HPO setup but are now coupled with tighter logging and evaluation routines — including offline inspection via `TensorBoard` — to facilitate more granular analysis of performance trends across the Top-14 configurations selected from Tables V and VI.

[5]https://github.com/StefanoGiacomelli/StrumKANet

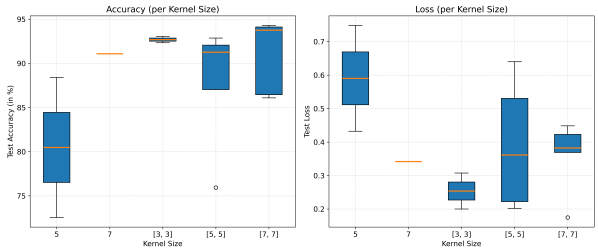

Fig. 11. Accuracy and Loss Distribution grouped by Kernel Size.

Figure 11 highlights the role of kernel size in shaping performance outcomes. Larger 2D kernels (e.g.: $[7, 7]$ and $[3, 3]$) consistently yield superior results — median accuracy above 92% and test loss between 0.2 and 0.4 — whereas 1D kernels show lower and more variable scores. These findings confirm that broader temporal–spectral context windows enhance the convolutional encoder's capacity to model rhythmic structure effectively, reinforcing the trends observed in the NAS-HPO phase.

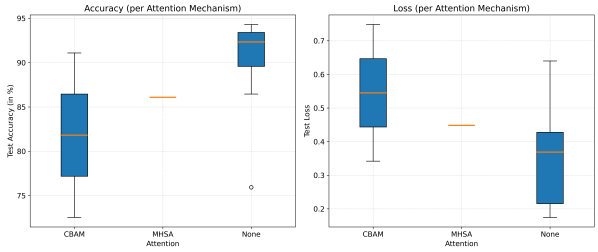

Fig. 12. Accuracy and Loss Distribution grouped by Attention Mechanism.

Figure 12 further validates the performance advantage of models without attention mechanisms, which achieve the highest median accuracy ($\sim 93\%$) and lowest test loss (median ~ 0.38). This confirms the limited utility of attention modules (CBAM, MHSA) observed during NAS-HPO. Even under full optimization, these mechanisms fail to consistently outperform simpler alternatives, suggesting that their contribution is marginal when deeper convergence is achieved.

Figure 13 shows that input features RP and OS again lead to the highest median accuracy (above 90%), reaffirming their discriminative value for strumming pattern classification. In contrast, RH-based models remain weaker and less stable. Models trained on Tempogram Ratios show broader loss and accuracy dispersion, indicating that their abstraction may limit generalization despite their temporal relevance.

Overall, these observations indicate that top-performing architectures under full training closely match those from the NAS-HPO stage. However, configurations involving attention modules or less informative features — although

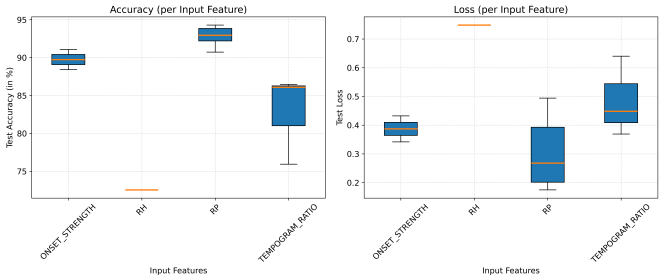

Fig. 13. Accuracy and Loss Distribution grouped by Input Features.

initially promising — often fail to maintain performance under extended optimization, suggesting issues like overfitting or reduced robustness.

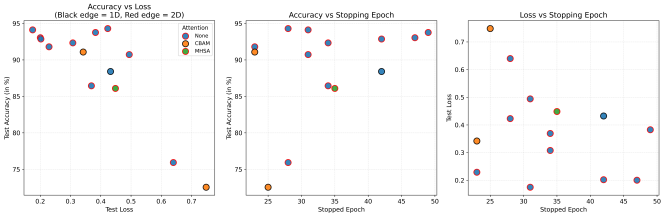

Fig. 14. Test accuracy and loss progression with respect to *early stopping epoch*. Attention and encoder type are visually encoded as in Figure 7.

Figure 14 analyzes the relationship between training duration (early stopping epoch) and final performance. The left panel reiterates the inverse correlation between loss and accuracy. The center and right panels show that early stopping time does not reliably predict better outcomes: high-accuracy models often converge before 30 epochs, particularly those with 2D kernels and no attention, while models requiring extended training often fail to generalize, despite reaching convergence. These results support two key insights: *(I)* attention mechanisms and smaller kernels increase variability in convergence behavior when used with WavKAN classifiers; and *(II)* high-performing models tend to converge earlier and more consistently, even in the presence of known KAN-related training latency, especially when optimized on expressive input features such as RP or OS.

VI. Conclusions

This paper presented a novel framework for classifying guitar strumming patterns in pop-rock songs, combining convolutional-attentive encoders with WavKANs. The proposed approach targets a mid-level MIR task — symbolic rhythmic classification — centered on gestures that may reflect either explicit strumming activity or perceptual pulsation (*Tactus*).

A large-scale NAS-HPO campaign demonstrated the effectiveness of rhythm-informed 1D features (Rhythm Patterns and Onset Strength) and wide convolutional kernels, with top models achieving up to $\sim 94\%$ accuracy. WavKANs, in particular, offered stable training and competitive performance, highlighting their suitability as interpretable approximators in music-driven tasks.

Beyond accuracy, results offer insights into how spectral–temporal descriptors relate to perceptual rhythm abstraction. In popular music, rhythmic hierarchies are intuitively internalized; novice guitarists rely on these cues to select appropriate strumming patterns. Our system emulates this mechanism by learning to infer rhythmic categories from pulse distributions, driven by beat-induced periodicities. The proposed class-aware smoothing further supports this abstraction, enhancing robustness to ambiguous meter boundaries.

Future works will extend the framework to tempo estimation, chord recognition, and strumming repetition tracking, enabling a full transcription pipeline from audio to chord–rhythm notation and bridging signal-level features with concrete and intelligent pedagogical applications.

References

[1] L. Turchet et al., "The internet of sounds: Convergent trends, insights, and future directions," *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11 264–11 292, 2023.

[2] S. Giacomelli, M. Giordano, and C. Rinaldi, "The ocon model: An old but green solution for distributable supervised classification for acoustic monitoring in smart cities," in *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)*, 2024, pp. 1–10.

[3] S. Giacomelli et al., "From large-scale audio tagging to real-time explainable emergency vehicle sirens detection," 2025. [Online]. Available: https://arxiv.org/abs/2506.23437

[4] F. Martusciello, C. Centofanti, C. Rinaldi, and A. Marotta, "Edge-Enabled Spatial Audio Service: Implementation and Performance Analysis on a MEC 5G Infrastructure," in *2023 4th International Symposium on the Internet of Sounds*. Pisa, Italy: IEEE, Oct. 2023, pp. 1–8.

[5] A. Chamberlain, A. Hazzard, E. Kelly *et al.*, "From AI, Creativity and Music to IoT, HCI, Musical Instrument Design and Audio Interaction: A Journey in Sound," *Personal and Ubiquitous Computing*, vol. 25, pp. 617–620, 2021.

[6] M. Pennese, S. Giacomelli, and C. Rinaldi, "The strummin' dataset: an international pop/rock curated audio selection for strumming patterns recognition," Jul. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15862786

[7] M. F. McKinney and D. Moelants, "Mirex 2006: Audio tempo extraction," https://www.music-ir.org/mirex/wiki/2006:Audio_Tempo_Extraction, 2006.

[8] P. Knees et al., "Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections," in *Proc. of the International Conference of the Internation Society for Music Information Retrieval (ISMIR)*, 2015.

[9] C. A. Harte, "Towards automatic extraction of harmony information from music signals," Ph.D. dissertation, Queen Mary, University of London, 2010. [Online]. Available: https://qmro.qmul.ac.uk/xmlui/handle/123456789/534

[10] Q. Xi et al., "Guitarset: A dataset for guitar transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[11] O. Nieto et al., "The harmonix set: Beats, downbeats, and functional segment annotations of western popular music," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[12] J. Glover, V. Lazzarini, and J. Timoney, "Real-time detection of musical onsets with linear prediction and sinusoidal modeling," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 68, 2011.

[13] K. Perlak, *Berklee Beginning Guitar*. Berklee Press, 2024, available from Berklee Press Guitar Series. [Online]. Available: https://berkleepress.com/music/guitar/

[14] A. Srinivasamurthy and X. Serra, "A supervised approach to hierarchical metrical cycle tracking from audio music recordings," in *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

TABLE VII
REFERENCE BASELINE FOR THE STRUMMIN' DATASET (WAVKAN W. CONVOLUTIONAL ATTENTIVE ENCODERS)

Encoder Params	Attention	Feature	Smoothing	Accuracy	Loss	Val Acc.	Stopping Epoch	Training Time
chs:32 ks:[7, 7]	**None**	**RP**	**True**	**0.9430**	**0.4230**	**0.7125**	**28**	**30m:49s**
chs:32 ks:[7, 7]	**None**	**RP**	**False**	**0.9412**	**0.1748**	**0.7196**	**31**	**34m:42s**
chs:16 ks:[7, 7]	None	RP	True	0.9376	0.3826	0.7125	49	44m:32s
chs:32 ks:[3, 3]	None	RP	False	0.9305	0.2001	0.6839	47	43m:20s
chs:32 ks:[5, 5]	None	RP	False	0.9287	0.2021	0.6857	42	40m:13s
chs:16 ks:[3, 3]	None	RP	False	0.9234	0.3076	0.6911	34	35m:02s
chs:16 ks:[5, 5]	None	RP	False	0.9180	0.2289	0.7036	23	27m:11s
chs:16 ks:[7]	CBAM	OS	False	0.9109	0.3419	0.4982	23	29m:35s
chs:16 ks:[5, 5]	None	RP	True	0.9073	0.4942	0.6911	31	32m:10s

Over-90% accuracy models only, in **Bold**: best configurations (Top-Accuracy and Min-Loss).

[15] Leonardo Nunes et al., "Beat and downbeat tracking based on rhythmic patterns applied to the uruguayan candombe drumming," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[16] U. Marchand and G. Peeters, "Scale and shift invariant time/frequency representation using auditory statistics: Application to rhythm description," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2016.

[17] H. Schreiber and M. Müller, "A Crowdsourced Experiment for Tempo Estimation of Electronic Dance Music," in *Proc. of the International Conference of the Internation Society for Music Information Retrieval (ISMIR)*, 2018.

[18] J. Gillick et al., "Learning to groove with inverse sequence transformations," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[19] T. Bertin-Mahieux et al., "The million song dataset," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[20] A. Correya et al., "Essentia.js: A JavaScript Library for Music and Audio Analysis on the Web," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*. Montréal, Canada: ISMIR, October 11–16 2020, pp. 605–612. [Online]. Available: http://hdl.handle.net/10230/45451

[21] ——, "Audio and music analysis on the web using essentia.js," *Transactions of the International Society for Music Information Retrieval*, Nov 2021. [Online]. Available: https://transactions.ismir.net/articles/10.5334/tismir.111

[22] S. Giacomelli et al., "Audioset-Tools: A python framework for taxonomy-aware audioset curation and reproducible audio research," *pre-print on Research Square, under peer-review for EURASIP - Journal on Audio Speech and Music Processing*, pp. 1–40, 2025.

[23] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Dec. 2019. [Online]. Available: http://arxiv.org/abs/1912.01703

[24] Y. Yang et al., "TorchAudio: Building Blocks for Audio and Speech Processing," Feb. 2022. [Online]. Available: http://arxiv.org/abs/2110.15018

[25] J. Hwang et al., "TorchAudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for PyTorch," Oct. 2023. [Online]. Available: http://arxiv.org/abs/2310.17864

[26] W. Falcon et al., "PyTorch Lightning," May 2020. [Online]. Available: https://zenodo.org/records/3828935

[27] B. McFee et al., "librosa: Audio and Music Signal Analysis in Python," in *Proceedings of the 14th Python in Science Conference*, Kathryn Huff and James Bergstra, Eds., 2015, pp. 18 – 24. [Online]. Available: https://proceedings.scipy.org/articles/Majora-7b98e3ed-003

[28] D. Bogdanov et al., "Essentia: an open-source library for sound and music analysis," in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 855–858. [Online]. Available: https://doi.org/10.1145/2502081.2502229

[29] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification," in *Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005)*, London, UK, September 11-15 2005, pp. 34–41. [Online]. Available: https://ismir2005.ismir.net/proceedings/1033.pdf

[30] S. Böck and G. Widmer, "Maximum Filter Vibrato Suppression for Onset Detection," in *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, 2013, pp. 55–61. [Online]. Available: https://www.dafx.de/paper-archive/details/0oee-99Z88WL7pSo749gcA

[31] P. Grosche, M. Müller, and F. Kurth, "Cyclic tempogram—a mid-level tempo representation for musicsignals," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 5522–5525.

[32] G. Peeters, "Rhythm classification using spectral rhythm patterns." in *ISMIR*, 2005, pp. 644–647. [Online]. Available: http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2005_ISMIR_RhythmClassification.pdf

[33] D. P. W. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007. [Online]. Available: https://www.ee.columbia.edu/~dpwe/pubs/Ellis07-beattrack.pdf

[34] R. Yu, W. Yu, and X. Wang, "KAN or MLP: A Fairer Comparison," *ArXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2407.16674

[35] J. Zhang et al., "Kolmogorov-Arnold Fourier Networks," *ArXiv*, 2025. [Online]. Available: https://arxiv.org/abs/2502.06018

[36] A. Koudounas et al., "KAN You Hear Me? Exploring Kolmogorov–Arnold Networks for Spoken Language Understanding," *arXiv preprint arXiv:2505.20176*, 2025. [Online]. Available: https://arxiv.org/abs/2505.20176v1

[37] T. d. Phuong, L.-v. Hoang, and H. d. Tran, "Pushing the Performance of Synthetic Speech Detection with Kolmogorov-Arnold Networks and Self-Supervised Learning Models," *arXiv preprint arXiv:2506.14153*, 2025. [Online]. Available: https://arxiv.org/abs/2506.14153v1

[38] M. Talha et al., "Fusion of Multimodal Audio Data for Enhanced Speaker Identification Using Kolmogorov-Arnold Networks," in *2024 IEEE International Conference on Consumer Electronics (ICCE)*, 2024, pp. 1–6.

[39] J. Zheng, M. Cao, and C. Zhang, "ICKAN: A Deep Musical Instrument Classification Model Incorporating Kolmogorov-Arnold Network," *Scientific Reports*, vol. 15, no. 21573, 2025. [Online]. Available: https://doi.org/10.1038/s41598-025-09493-y

[40] C. Sun et al., "Learning high-frequency functions made easy with sinusoidal positional encoding," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[41] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[42] S. Woo et al., "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[43] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[44] J. Ho et al., "Axial attention in multidimensional transformers," *arXiv*, 2019. [Online]. Available: https://arxiv.org/abs/1912.12180

[45] Z. Bozorgasl and H. Chen, "Wav-kan: Wavelet kolmogorov-arnold networks," 2024. [Online]. Available: https://arxiv.org/abs/2405.12832

[46] Z. Liu et al., "Kan: Kolmogorov-arnold networks," 2025. [Online]. Available: https://arxiv.org/abs/2404.19756

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980