JOINT ESTIMATION OF PIANO DYNAMICS AND METRICAL STRUCTURE WITH A MULTI-TASK MULTI-SCALE NETWORK

Zhanhong He[1], Hanyu Meng[2], Defeng (David) Huang[1], Roberto Togneri[1]

[1]The University of Western Australia, Perth, Australia
[2]The University of New South Wales, Sydney, Australia

ABSTRACT

Estimating piano dynamic from audio recordings is a fundamental challenge in computational music analysis. In this paper, we propose an efficient multi-task network that jointly predicts dynamic levels, change points, beats, and downbeats from a shared latent representation. These four targets form the metrical structure of dynamics in the music score. Inspired by recent vocal dynamic research, we use a multi-scale network as the backbone, which takes Bark-scale specific loudness as the input feature. Compared to log-Mel as input, this reduces model size from 14.7 M to 0.5 M, enabling long sequential input. We use a 60-second audio length in audio segmentation, which doubled the length of beat tracking commonly used. Evaluated on the public MazurkaBL dataset, our model achieves state-of-the-art results across all tasks. This work sets a new benchmark for piano dynamic estimation and delivers a powerful and compact tool, paving the way for large-scale, resource-efficient analysis of musical expression.

Index Terms— Piano dynamics estimation, beat tracking, Bark-scale specific loudness, multi-scale network, multi-task learning

1. INTRODUCTION

The creation, comprehension and reproduction of music are fundamental aspects of human culture. In Western musical tradition, the term *dynamics* refers to a coarse guide to the intended loudness. Indicated in a score by markings such as *p* (*piano*, soft) and *f* (*forte*, loud), dynamics are essential for shaping musical phrases, conveying emotion, and articulating structural differentiation [1]. This vocabulary extends to a nuanced range of static levels, from *pp* (*pianissimo*) to *ff* (*fortissimo*), and includes gradual transitions like *crescendo* and *decrescendo*. The computational modeling of these expressive markings is valuable for music education and performance analysis [2–4], as well as theory-informed music generation [5].

The core challenge in estimating from audio lies in the inherent relativity and ambiguity of dynamic markings. A symbol like *pianissimo* does not map to a fixed physical measurement like a decibel level. Instead, its interpretation is deeply contextual, influenced by musical style, performer's intent, and the acoustic environment [6,7]. This lack of a standardized ground truth has historically posed a significant challenge for machine learning algorithms, often leading to poor generalization among different performers or pieces [8,9].

To circumvent the ambiguity of music dynamics, a common strategy in music transcription and analysis is to adopt MIDI velocity as a proxy target [10–13]. This approach, however, introduces its own set of challenges. MIDI velocity reflects the performer's physical action rather than perceived loudness, and is confounded by the instrument's unique timbre and touch [14]. While automatic music transcription (AMT) systems can accurately estimate MIDI velocity from Yamaha Disklavier piano performances, generalizing this capability across diverse pianos remains unsolved [15]. This makes the subsequent task of regressing dynamic markings from the estimated MIDI velocity inherently unreliable.

Given these complexities, we propose an end-to-end multi-task learning approach to estimate piano dynamics and their underlying metrical structure directly from audio. Inspired by recent advances in vocal dynamics estimation [16], we use Bark-scale specific loudness (BSSL) as the input feature. The BSSL is processed by a multi-scale network backbone, adapted from [17], to extract a shared latent representation that reconciles the divergent temporal requirements of the distinct tasks. This unified latent representation is then channeled through a Multi-gate Mixture-of-Experts (MMoE) layer [18], which generates specialized features for four task heads that jointly predict: (1) dynamic levels, (2) change points, (3) beat positions, and (4) downbeat positions. Together, these targets capture both the dynamic markings and their underlying metrical structure, since the beat and downbeat grid provides the rhythmic foundation of a musical score.

The primary application of our proposed multi-task framework is to enrich musical scores that possess reliable beat information but lack dynamic markings, a common scenario for music archives and the output of score-level AMT systems [19]. As depicted in Fig. 1, this workflow aligns the model's predicted dynamics to a provided beat grid. Furthermore, the model's ability to jointly predict the beat and downbeat also allows it to operate fully end-to-end without a pre-existing music score, making it a versatile tool for large-scale piano performance analysis directly from audio.[1]

2. METHODOLOGY

2.1. Bark-Scale Specific Loudness

Bark-scale features are derived from a psychoacoustic model of human loudness perception across critical bands [20]. While log-Mel spectrograms remain dominant in modern deep learning pipelines, the effectiveness of Bark-based representations is supported by extensive research. For instance, recent work has demonstrated that Bark-scale cepstral coefficients can improve speaker recognition under short-duration constraints [21], and a high-resolution variant of BSSL has shown performance gains over log-Mel inputs for vocal dynamics estimation [16]. Given the growing evidence supporting Bark-scale features, coupled with BSSL's established success in previous piano dynamics research [8], we adopt it as the foundational feature for this work.

---

[1]Code and pre-trained models are available at: `https://github.com/zhanh-he/piano-dynamic-estimation`
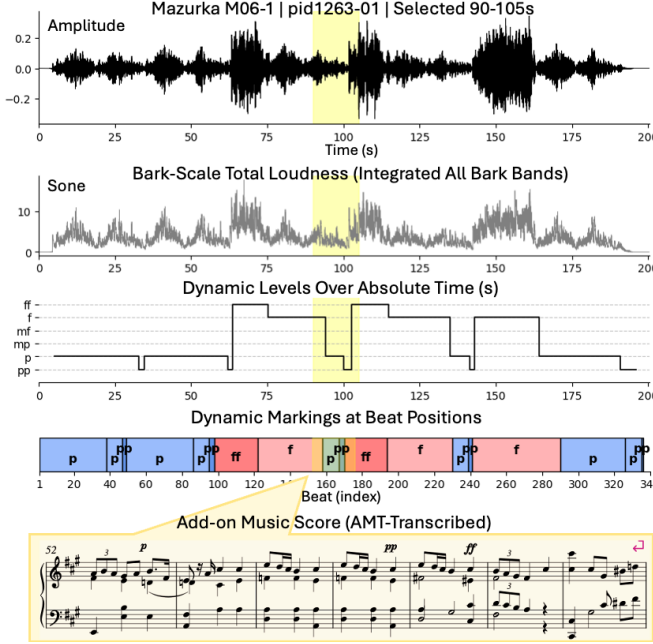
Fig. 1. Workflow for adding dynamic markings to an AMT-transcribed score from audio. The highlighted region (90–105 s of Chopin Mazurka Op. 6 No. 1) shows a case study. Dynamic markings are aligned with the musical beat grid, and a change point indicates a contextual shift in the dynamic levels.

The primary challenge was the implementation of the BSSL extractor. Standardized toolboxes like MOSQITO [22], while compliant with ISO 532-1:2017 standard, are designed for robust sound quality assessment and thus mandate a 48 kHz sampling rate with constrained STFT parameters. Since most music and piano recordings are encoded at 44.1 kHz, using MOSQITO would necessitate upsampling, a computationally expensive process that can introduce interpolation artifacts. Therefore, we developed a custom BSSL feature extractor in the PyTorch framework, reproducing the `ma_sone` function from the widely used Music Analysis MATLAB toolbox by Pampalk [23], which is based on the algorithmic chain proposed in [24]. This approach allows for the flexible parameter settings crucial for fair and efficient experimentation.

The feature extraction pipeline begins by converting the audio to mono, peak-normalizing it to -1 dBFS. To ensure compatibility with a recent state-of-the-art (SOTA) beat tracking model [25], the audio is resampled to 22.05 kHz, and a short-time Fourier transform (STFT) is computed using a Hann window of length 1024 and a frame rate of 50 fps (i.e., 20 ms per frame and hop size of 441). The resulting power spectrogram is then transformed into the BSSL through a series of psychoacoustic modeling steps. These include outer- and middle-ear weighting, grouping spectral energy into critical bands, modeling spectral masking, and nonlinear mapping to the perceptual *sone* scale. This process yields the final input feature, a BSSL matrix $\mathbf{X} \in \mathbb{R}^{22 \times T}$, where T denotes the number of time frames. The choice of 22 Bark bands is determined by the 11.025 kHz Nyquist frequency, which fully covers the 22nd Bark band (extending to 9.5 kHz). While the total loudness provides an intuitive visualization (Fig.1), the $22 \times T$ BSSL matrix is input to the model to preserve rich spectral details.

2.2. Model Architecture

The proposed model architecture is illustrated in Fig. 2. With divergent acoustic requirements, dynamic estimation requires a large temporal receptive field [16], whereas beat and downbeat tracking require high temporal resolution to locate transient onsets [26]. To address this need for varied receptive fields, we adapt the multi-scale network from [17] as a shared encoder. Given an $F \times T$ time–frequency feature as input, the features are first normalized via 2D Batch Normalization (requiring a dimension transposition). The encoder's branches are then built from a series of residual and self-attention blocks detailed in [17]. The encoder operates on three such parallel branches at different temporal resolutions, corresponding to lengths of T, T/s, and T/s^2, where down- and upsampling by the scaling factor s are achieved using strides in max-pooling and transposed convolutions. Our architecture departs from the original design [17] in two key ways: we treat the scaling factor s as a configurable hyperparameter rather than a fixed value of 3, and the encoder outputs a shared latent sequence $\mathbf{Z} \in \mathbb{R}^{T \times 8}$ for later processing, rather than feeding into a single-task classifier.

To mitigate negative transfer between our acoustically diverse tasks, we develop a MMoE module [18] as a task-aware decoder that processes the shared representation. The module consists of two main components:

- **Shared Experts**: The module contains a pool of 8 shared experts. Each expert in $i = 1, \ldots, 8$ is a lightweight temporal convolutional block. It comprises two sequential 1D convolutional layers, each with a kernel size of 3 and a length-preserving padding, separated by a ReLU activation. All experts process the shared latent sequence $\mathbf{Z}$ in parallel to produce a set of expert outputs.
- **Task-Specific Gates**: For each of the four tasks k, a dedicated gating network $G_k(\cdot)$ acts as a dynamic router. Each gate is implemented as a simple linear layer that takes the latent feature $\mathbf{z}_t$ at each time step t and outputs a softmax-normalized weight vector $\mathbf{w}_k(t) \in \mathbb{R}^8$. This vector determines how to weigh the contributions of the different experts for that specific task.

The final feature vector for task k at a given time step t, denoted $\mathbf{y}_{k,t}$, is computed as the weighted sum of gates with all expert outputs:

$$\mathbf{y}_{k,t} = \sum_{i=1}^{8} \mathbf{w}_{k,i}(t) \cdot \mathbf{e}_{i,t} \quad \text{where} \quad \mathbf{w}_k(t) = \mathrm{Softmax}(G_k(\mathbf{z}_t)) \quad (1)$$

where $\mathbf{e}_{i,t}$ is the output of the i-th expert, and the weight vector $\mathbf{w}_{k,i}(t)$ is produced by the task's gate, and $\mathbf{w}_{k,i}(t)$ denotes its i-th component, representing the importance of expert i for task k at that time step. This computation is performed at each frame to form the complete task-specific feature $\mathbf{Y}_k = [\mathbf{y}_{k,1}, \ldots, \mathbf{y}_{k,T}]^\top \in \mathbb{R}^{T \times 8}$. These four specialized representations are then mapped to the final logits by separate linear heads.

2.3. Task-Specific Post-Processing

The model's raw output is a sequence of frame-wise probabilities (i.e., logits), which we convert into discrete musical events via tailored post-processing. For beat and downbeat detection, we adopt the procedure from [25], identifying events by thresholding probabilities at 50% and applying peak-picking within a 70 ms neighborhood (± 3 frames at 50 fps). For dynamic markings, i.e., the dynamic level of each detected beat is determined by taking the argmax of the class probabilities at that specific time frame. Finally, change points are determined in a two-step process: we first identify all frames where the probabilities exceed a 75% threshold, and then snap each
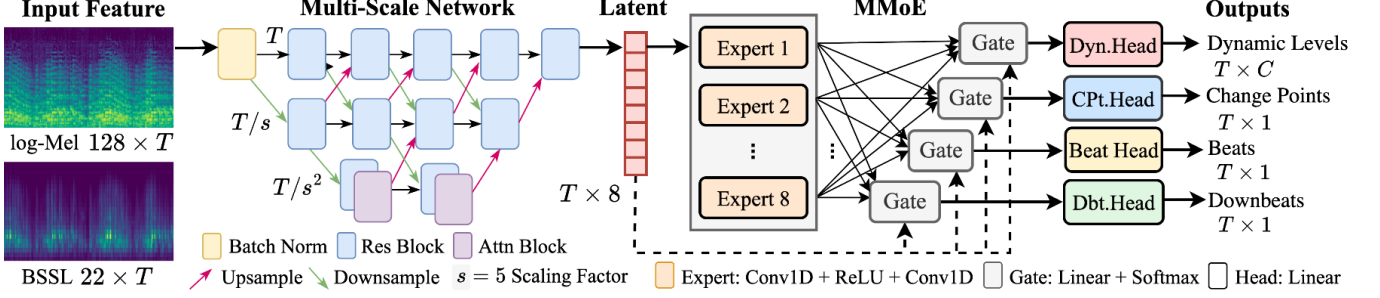
Fig. 2. Architecture of the proposed multi-task framework. A three-branch multi-scale network encodes either log-Mel or BSSL input. Branches operate at lengths T, T/s, and T/s^2, with outputs a latent sequence of shape $T \times 8$. An 8-expert MMoE with four gates forms task-specific features, followed by linear heads to output four distinct targets. The scaling factor s is a tunable hyperparameter.

of these candidates to the nearest detected beat to determine its location. This final beat-snapping step is performed because, while not a strict musical rule, the annotations (e.g., dynamic markings) in the MazurkaBL dataset are exclusively beat-aligned [9], and this approach maintains consistency with prior work [27].

2.4. Loss Function

Training the multi-task model involves a composite loss that combines specialized objectives for each task. Our multi-task loss, $\mathcal{L}_{\mathrm{MTL}}$, is defined as the sum of four task-specific losses:

$$\mathcal{L}_{\mathrm{MTL}} = \mathcal{L}_{\mathrm{Dyn}} + \mathcal{L}_{\mathrm{CPt}} + \mathcal{L}_{\mathrm{Beat}} + \mathcal{L}_{\mathrm{Dbt}} \tag{2}$$

where each component is defined as follows. For the binary targets (change points, beats, and downbeats), the loss terms $\mathcal{L}_{CPt}$, $\mathcal{L}_{Beat}$, and $\mathcal{L}_{Dbt}$ employ the shift-tolerant weighted binary cross-entropy proposed in [25]. This function addresses two key challenges: it counteracts the extreme sparsity of targets by weighting positive frames more heavily, and it accommodates annotation timing imprecision by incorporating a ± 3 frame tolerance window. For the multiclass target, dynamics, the loss term $\mathcal{L}_{Dyn}$ is a standard cross-entropy, masked by the ground-truth beat positions. This marking enforces a data-driven prior on the dataset's annotation style (i.e., dynamic markings occur only on beats), guiding the model to ignore spurious inter-beat fluctuations.

3. EXPERIMENTS

3.1. Dataset

We use the MazurkaBL dataset [9], a score-aligned corpus of 2,098 solo piano recordings covering 44 Chopin Mazurkas. After excluding two mazurkas with irregular dynamics annotations (M06-4 and M63-2), 1,999 recordings are used. While similar datasets like the recently introduced EME33 [28] exist, MazurkaBL is the largest publicly available resource for studying notated dynamics versus performed loudness from audio. Its provision of score-aligned beat times and verified expressive markings has led to its wide adoption in music analysis research. Designed for different goals, other large datasets like MAESTRO [29] use MIDI velocity as a proxy for dynamics, while others such as ASAP [30] provide score-aligned annotations but lack dynamic markings.

3.2. Implementation Details

We use a 5-fold cross-validation protocol, stratified by 44 mazurkas, to train and evaluate our model. For each fold, the model is trained for 120 epochs, with the best-performing checkpoint selected based on the F1 score on its respective validation set. The training configuration includes the AdamW optimizer with a learning rate of 3e-4, a batch size of 10, and a fixed random seed of 86.

For data augmentation, we slice audio into 60-second segments with 50% overlap during training, while no overlap is used for evaluation. Both BSSL (22 Bark bands) and log-Mel (128 mel bins) features are extracted using the same STFT parameters as in [25] to ensure identical temporal resolution. The model employs a compact configuration with an empirically optimal temporal scaling factor of $s = 5$. It predicts dynamic levels across $C = 6$ classes: a *blank* class for silence before the first annotation, and the five dynamic classes (*pp, p, mf, f, ff*) from the MazurkaBL dataset. When run on a NVIDIA RTX 3090 (24 GiB), a full 5-fold run completed in approximately 20 hours, with peak GPU memory usage of 4 GiB.

3.3. Baselines

We compare the proposed multi-task network against the single-task network and task-specific baselines.
Single-task Multi-scale Network. We establish single-task baselines by adapting the multi-scale network from [17], integrating their model architecture (code publicly available) into our data handling pipeline and training an independent model for each target. The loss for each single-task network is the corresponding single term from our multi-task loss, with an empirically optimal temporal scaling factor of 5.
Dynamic and Change Point Baselines. We report published results from representative methods in [27]. This includes the Artificial Neural Networks (ANN) for dynamics, and the Pruned Exact Linear Time (PELT) algorithms for change points. These methods are tuning-intensive and non-end-to-end, so we report the literature scores rather than re-implement them.
Beat and Downbeat Baselines. We include the time-convolutional network (TCN) with a dynamic Bayesian network (DBN) post-processor [26], and the recent state-of-the-art transformer model, Beat This [25]. Both models can estimate beats and downbeats simultaneously. We retrain them from scratch on the MazurkaBL dataset using their publicly available code and the same 5-fold protocol as ours.

Table 1. Performance comparison of the proposed model against all baselines. Per-task F1 scores (%) are reported as mean ± standard deviations over 5-fold cross-validation. PELT is an algorithmic method with no trainable parameters, while the ANN did not report this attribute. The best score is highlighted in bold.

Method	Feature	Dynamic F1	Change Point F1	Beat F1	Downbeat F1	# Params
ANN [27]	BSSL	29.4	–	–	–	n/a
PELT [27]	BSSL	–	10.8	–	–	n/a
TCN+DBN [26]	log-Mel	–	–	60.9 ± 1.8	30.4 ± 1.3	0.1 M
Beat This [25]	log-Mel	–	–	80.5 ± 2.7	52.8 ± 6.2	20.3 M
Single-task Multi-scale Network	BSSL	50.6 ± 10.1	21.0 ± 9.9	84.0 ± 1.5	45.0 ± 1.7	0.4 M
w/o. BSSL	log-Mel	50.4 ± 11.1	17.5 ± 5.4	83.8 ± 1.8	54.7 ± 7.5	13.3 M
Multi-task Multi-scale Network (Proposed)	BSSL	$\mathbf{54.4 \pm 8.9}$	$\mathbf{26.1 \pm 9.7}$	$\mathbf{84.1 \pm 1.3}$	55.2 ± 4.2	0.5 M
w/o. BSSL	log-Mel	50.8 ± 10.9	23.1 ± 6.1	83.7 ± 1.7	$\mathbf{58.5 \pm 6.2}$	14.7 M

3.4. Evaluation Metrics

Performance on all four tasks is evaluated using the F1 score. For beat and downbeat tracking, we report F1 with a ±70 ms tolerance, consistent with prior work [25, 26]. Evaluation of both dynamics and change points is consistent with [27]. For dynamics, the model's continuous output (dynamic level curve) is first sampled at each ground-truth beat location, and these values are then discretized into the corresponding dynamic markings. This converts the frame-wise prediction into a sequence of beat-wise labels, which are evaluated using a macro-averaged F1 score across five dynamic classes (*pp, p, mf, f, ff*, excluding the *blank* class). For change points, their predictions are snapped to the nearest ground-truth beat before being evaluated with a standard F1 score. This beat-wise alignment for both tasks turns the evaluation into a direct, musically metrical index-based comparison, thus requiring no additional timing tolerance.

4. RESULTS

4.1. Main Result

As presented in Table 1, our proposed multi-task model achieves SOTA performance in dynamics and change point estimation, while performing competitively on the remaining tasks. The effectiveness of the multi-task learning paradigm is underscored by the model's superior performance relative to its single-task counterpart using the same BSSL features. This includes significant F1 score improvements in dynamics (+3.8%), change points (+5.1%), beats (+0.1%), and downbeats (+10.2%). Beyond these quantitative improvements, the multi-task model offers considerable practical utility. It operates within a highly parameter-efficient framework (0.5 M vs. 4× single-task 0.4 M) and can utilize its own predicted beat positions for post-processing, enabling practical application on unannotated audio.

Our analysis also reveals a strong task-feature dependency: BSSL features are optimal for dynamics, change points, and beats estimation, whereas log-Mel features are preferable for downbeat tracking. A primary advantage of BSSL is its compactness (22 Bark bins vs. 128 Mel bins in our STFT setup). Within our multi-task multi-scale network, which relies on convolutional residual blocks, this smaller input dimension can reduce the model's trainable parameters from 14.7 M to just 0.5 M. This significantly smaller footprint enables the model to process longer audio sequences, directly benefiting tasks that require long-term temporal information and highlighting BSSL's potential for a wider range of musical applications with long-term dependencies.

Table 2. Ablation study of the multitask network with BSSL features. Per-task F1 scores (%, mean only) and their average are reported over 5-fold, showing impacts of disabling key components.

Setting	Dyn F1	CPt F1	Bt F1	Dbt F1	Average
Proposed	**54.4**	**26.1**	**84.1**	**55.2**	**55.0**
w/o. MMoE	52.8	22.0	82.9	51.8	52.4
w/o. Temp. Scal.	50.5	13.3	80.3	41.9	46.5
w/o. Data Augm.	50.5	19.6	83.2	51.7	51.2
uses 30s Segment	49.1	19.2	83.4	52.7	51.1

4.2. Ablation Study

To validate our design choices, we conduct a comprehensive ablation study, introducing an average score (calculated by averaging the 5-fold mean F1 scores from the four tasks) to measure global performance. We systematically evaluate four configurations against our full model: (i) removing the MMoE module; (ii) disabling the multi-scale functionality by setting the scaling factor $s = 1$; (iii) removing data augmentation by using non-overlapping 60-second audio segments in training stage; and (iv) reducing the input audio length from our default setting 60-second to 30-second (same length as in [25]). As detailed in Table 2, each of the proposed components and training choices contributes meaningfully to the model's final performance, with the extended 60-second input context providing a significant advantage in dynamics-related tasks.

5. CONCLUSION

In this paper, we proposed a compact multi-task, multi-scale network that jointly estimates piano dynamics, change points, beats, and downbeats directly from audio. Using Bark-scale specific loudness as input and an MMoE decoder, our model leverages a 60-second temporal context while remaining highly parameter-efficient with only 0.5 M parameters. Evaluated on the MazurkaBL dataset, our model achieves state-of-the-art results for dynamics and change point detection, while demonstrating competitive performance in beat and downbeat tracking. This demonstrates not only the model's practical utility but also its significant potential for broader applications. Future work will focus on combining our proposed model with score-level piano transcription systems. Such an end-to-end pipeline could produce music scores with dynamic markings from the performance audio, but developing appropriate evaluation methods for such comprehensive outputs presents a new challenge.

6. REFERENCES

[1] Carlos E. Cancino-Chacón, Maarten Grachten, Werner Goebl, and Gerhard Widmer, "Computational models of expressive music performance: A comprehensive and critical review," *Frontiers in Digital Humanities*, vol. 5, pp. 25, 2018.

[2] Jyoti Narang, Marius Miron, Ajay Srinivasamurthy, and Xavier Serra, "Analysis of musical dynamics in vocal performances using loudness measures," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Vienna, Austria, 2022, pp. 301–311.

[3] Huan Zhang, Vincent K.M. Cheung, Hayato Nishioka, Simon Dixon, and Shinichi Furuya, "Llaqo: Towards a query-based coach in expressive music performance assessment," in *Proc. ICASSP*, 2025, pp. 1–5.

[4] Eun Ji Park, "Music dynamics visualization for music practice and education," *Multimedia Tools and Applications*, vol. 84, no. 49, pp. 36145–36161, 2025.

[5] Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 2692–2703, 2024.

[6] Katerina Kosta, Oscar F. Bandtlow, and Elaine Chew, "Practical implications of dynamic markings in the score: Is piano always piano?," in *AES 53rd International Conference: Semantic Audio*, 2014, pp. 1–3.

[7] Gabriel Jones and Anders Friberg, "Probing the underlying principles of dynamics in piano performances using a modelling approach," *Frontiers in Psychology*, vol. 14, pp. 1269715, 2023.

[8] Katerina Kosta, Rafael Ramirez, Oscar F. Bandtlow, and Elaine Chew, "Mapping between dynamic markings and performed loudness: A machine learning approach," *Journal of Mathematics and Music*, vol. 10, no. 2, pp. 149–172, 2016.

[9] Katerina Kosta, Oscar F. Bandtlow, and Elaine Chew, "Mazurkabl: Score-aligned loudness, beat, and expressive markings data for 2000 chopin mazurka recordings," in *Proc. Int. Conf. Technologies for Music Notation and Representation (TENOR)*, 2018, pp. 85–94.

[10] Sam van Herwaarden, Maarten Grachten, and W. Bas de Haas, "Predicting expressive dynamics in piano performances using neural networks," in *Proc. ISMIR*, 2014, pp. 47–52.

[11] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Sageev Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck, "Onsets and frames: Dual-objective piano transcription," in *Proc. ISMIR*, 2018, pp. 50–57.

[12] Hyesung Kim, Marius Miron, and Xavier Serra, "Score-informed midi velocity estimation for piano performance by film conditioning," in *Proc. of the Sound and Music Computing Conf. (SMC)*, 2023, pp. 139–147.

[13] Hyesung Kim and Xavier Serra, "A method for midi velocity estimation for piano performance by a u-net with attention and film," in *Proc. ISMIR*, 2024, pp. 304–310.

[14] Axel Berndt and Tilo Hähnel, "Modelling musical dynamics," in *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*, 2010, pp. 1–8.

[15] Drew Edwards, Simon Dixon, Emmanouil Benetos, Akira Maezawa, and Yuta Kusaka, "A data-driven analysis of robust automatic piano transcription," *IEEE Signal Processing Letters*, vol. 31, pp. 681–685, 2024.

[16] Jyoti Narang, Nazif Can Tamer, Viviana De La Vega, and Xavier Serra, "Automatic estimation of singing voice musical dynamics," in *Proc. ISMIR*, 2024, pp. 256–263.

[17] Dichucheng Li, Mingjin Che, Wenwu Meng, Yulun Wu, Yi Yu, Fan Xia, and Wei Li, "Frame-level multi-label playing technique detection using multi-scale network and self-attention mechanism," in *Proc. ICASSP*, 2023, pp. 1–5.

[18] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2018, pp. 1930–1939.

[19] Wei Zeng, Xian He, and Ye Wang, "End-to-end real-world polyphonic piano audio-to-score transcription with hierarchical decoding," in *Proc. IJCAI*, 2024, pp. 7788–7795.

[20] Eberhard Zwicker and Hugo Fastl, *Psychoacoustics: Facts and Models*, vol. 22 of *Springer Series in Information Sciences*, Springer, Berlin, 2nd, updated edition, 1999.

[21] Yong Zi and Jibin Xiong, "Improving short-duration speaker recognition by joint bark-wavelet acoustic feature coupling and pitch-weighted posterior encoding," *Wireless Personal Communications*, vol. 136, pp. 1111–1130, 2024.

[22] Roberto San Millán-Castillo, Eduardo Latorre-Iglesias, Martin Glesser, Salomé Wanty, Daniel Jiménez-Caminero, and José María Álvarez-Jimeno, "Mosqito: an open-source and free toolbox for sound quality metrics in the industry and education," in *Inter-Noise and NOISE-Con Congress and Conference Proceedings*, 2021, pp. 1164–1175.

[23] Elias Pampalk, "A matlab toolbox to compute music similarity from audio," in *Proc. ISMIR*, 2004, pp. 254–257.

[24] Elias Pampalk, Andreas Rauber, and Dieter Merkl, "Content-based organization and visualization of music archives," in *Proc. ACM Int. Conf. Multimedia*, 2002, pp. 570–579.

[25] Francesco Foscarin, Jan Schlüter, and Gerhard Widmer, "Beat this! accurate beat tracking without dbn postprocessing," in *Proc. ISMIR*, 2024, pp. 962–969.

[26] Sebastian Böck and Matthew E.P. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proc. ISMIR*, 2020, pp. 574–582.

[27] Katerina Kosta, *Computational Modelling and Quantitative Analysis of Dynamics in Performed Music*, Ph.D. thesis, Centre for Digital Music, Queen Mary University of London, 2017.

[28] Tzu-Ching Hung, Jingjing Tang, Kit Armstrong, Yi-Cheng Lin, and Yi-Wen Liu, "Eme33: A dataset of classical piano performances guided by expressive markings with application in music rendering," in *Proc. of IEEE Int. Conf. on Big Data (BigData)*, 2024, pp. 3174–3180.

[29] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck, "Enabling factorized piano music modeling and generation with the maestro dataset," in *International Conference on Learning Representations (ICLR)*, 2019.

[30] Francesco Foscarin, Andrew McLeod, Philippe Rigaux, Florent Jacquemard, and Masahiko Sakai, "ASAP: a dataset of aligned scores and performances for piano transcription," in *Proc. ISMIR*, 2020, pp. 534–541.