# Attention-based Adaptive Structured Patchout Spectrogram Transformer for Music Classification

Yuan Liu* Lingqing Liu* Yichen Yang* Shoji Makino*
* Waseda University, Japan
E-mail: yuan-liu@ruri.waseda.jp, lingqing_liu@moegi.waseda.jp, yang_yichen@mail.nwpu.edu.cn, s.makino@waseda.jp

*Abstract*—**Music classification is a subfield of audio classification, which aims to understand and differentiate various types of music. Patchout faSt Spectrogram Transformer (PaSST) is an efficient transformer model for audio classification. Its core innovation is Patchout, a token-dropping mechanism. Patchout reduces computational demands and improves model generalization by shortening the input sequence. However, its random patch selection strategy may lead to the loss of crucial information. This loss hinders full feature learning, especially for structurally rich music. Therefore, we proposed a novel Attention-based Adaptive Structured Patchout (AASPatchout). It deterministically preserves the most informative patches by using the transformer's internal attention scores. Our AASPatchout requires no additional modules, instead directly utilizing the model's existing attention mechanism. We evaluated AASPatchout for two downstream tasks in music classification: polyphonic instrument recognition and music genre classification. The experimental results demonstrated that our proposed method outperformed the conventional random Patchout baseline.**

## I. Introduction

Audio classification aims to classify audio segments into predefined categories in various acoustic scenarios [1]–[4]. As a distinct application in audio classification, music classification [2] plays a significant role aiming to classify music into various categories such as genre, artist, or mood. In particular, unlike ordinary audio, music exhibits rich structure, such as timbre, harmony, melody, and rhythm [5]. This complexity makes it difficult to classify music accurately and effectively.

Motivated by the great success of transformers in natural language processing [6], various attempts have been made to adapt transformers to other domains, such as vision [7] and audio. The Audio Spectrogram Transformer (AST) [8] was the first purely attention-based audio transformer model and outperformed Convolutional Neural Networks (CNNs) on audio classification. However, self-attention's complexity scales quadratically with sequence length. Consequently, transformers exhibit larger parameters, higher computational demands and greater memory overhead than CNNs. To solve this problem, various architectures have been proposed [9]–[13]. For example, Hierarchical Token-Semantic Audio Transformer (HTS-AT) employed hierarchical windows to progressively compress token sequences [9], achieving state-of-the-art (SOTA) results with fewer parameters. The Patchout technique was adopted into AST, leading to the Patchout faSt Spectrogram Transformer (PaSST) [10]. Compared to other optimization methods, a key advantage of PaSST is that its Patchout is simple while obtaining significant performance improvements. However, the conventional Patchout method still has limitations due to its random patch selection strategy. It may disrupt the melodic components of the musical signal that convey critical information. This loss hinders effective available data utilization and the discovery of key classification features.

To address the limitation, we proposed to transform random Patchout into deterministic Patchout. Notably, we observed that Patchout is highly related to token pruning [14], as both reduce the computational complexity of the transformer by removing input tokens. Among various token pruning methods, Expediting Vision Transformers (EViT) [15] is a notable example, which integrates token pruning techniques. It utilizes the transformer's attention mechanism to identify and retain image patches that contribute the most to classification tasks. Therefore, inspired by EViT's approach to identify patch importance, we proposed an Attention-based Adaptive Structured Patchout (AASPatchout). We evaluated our proposed method on OpenMIC-2018 [16] and GTZAN [17]. The results show that our method increases the mean average precision to 84.95 % and the accuracy to 86.11 %, outperforming the random Patchout baseline.

## II. Related Work

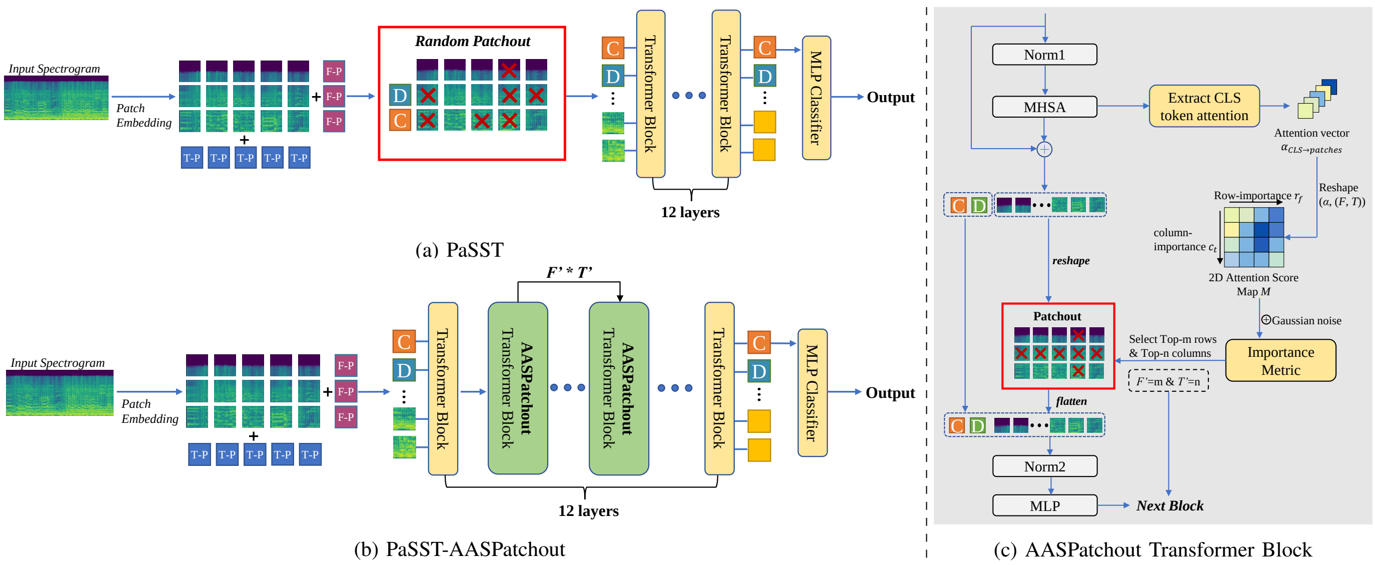
### A. Patchout faSt Spectrogram Transformer

PaSST is an efficient transformer architecture specifically designed for audio classification tasks and serves as our baseline model.

Instead of directly applying absolute positional encoding, PaSST disentangles the positional encoding for the time and frequency dimensions. This design enables audio to be split along the time dimension without changing the relationships of frequency positional encoding. This allows for the flexible processing of different lengths of audio. Importantly, this lays the foundation for the implementation of the Patchout mechanism, while it is also more suitable to the spectro-temporal structure of the audio spectrograms.

Patchout is the core mechanism of PaSST. During the training phase, Patchout randomly drops patches to shorten sequences. It takes two primary forms. Specifically, structured Patchout randomly picks some frequency bins/time frames and removes a whole column/row of patches, while unstructured Patchout randomly removes patches regardless of their position. Moreover, besides shortening the input sequence, Patchout also implements effective regularization, thereby improving the model's generalization and overall performance.

979-8-3315-7206-8/25/$31.00 ©2025 IEEE



(a) PaSST

(b) PaSST-AASPatchout

(c) AASPatchout Transformer Block

Fig. 1: Overall architecture comparison. (a) architecture of PaSST, (b) architecture of the proposed PaSST-AASPatchout. (c) detailed structure of AASPatchout Transformer Block

### B. Token Pruning

In CV, many image regions are irrelevant to classification tasks. Yet they still significantly impact efficiency and overhead due to the high computational demands of transformers. To address this redundancy, token pruning is proposed, which removes input tokens selectively [18]. Whether achieved through token merging or token dropping, its core principle is retaining key feature tokens. As a result, token pruning accelerates training and inference while maintaining performance.

### C. Expediting Vision Transformers via Token Reorganizations

EViT is a Vision Transformer (ViT) model integrating the token pruning technique, which has demonstrated excellent performance in CV. It points out that patches with higher attention scores are often more discriminative. Consequently, EViT defines the top k% of patches with the highest scores as attentive token, and the rest as inattentive token. Then, the attentive tokens are preserved, while inattentive tokens are fused to reduce computational overhead. Therefore, EViT reduces the number of tokens processed in deeper network layers, accelerating training and inference and lowering computational requirements.

The core principle of EViT provides a promising strategy for enhancing PaSST's Patchout mechanism. Specifically, this allows us to directly utilize the model's existing attention mechanisms to select patches, rather than requiring an auxiliary network or module for selection.

## III. Proposed Method

To address the limitation of conventional Patchout and improve the classification ability, we proposed an Attention-based Adaptive Structured Patchout (AASPatchout) using the transformer's existing attention scores to select patches. Furthermore, we introduced a progressive, multi-stage Patchout strategy. Through this design, the staged processing allows the model to retain the most relevant information at different depths.

### A. Workflow

Fig. 1 contrasts the workflows of (a) PaSST and (b) our proposed PaSST-AASPatchout model. First, Mel spectrograms are split into 16x16 patches and then linearly projected into patch embedding vectors. Next, these patches are combined with time positional encoding and frequency positional encoding. Since the importance recognition of patches requires attention scores, the Patchout stage is delayed and integrated into the transformer block. Therefore, after patches are input into the transformer encoder layers, the model drops patches in AASPatchout transformer block. Finally, output sequence from the 12 encoder layers is passed to the classification head for the final task.

### B. Attention-based Adaptive Structured Patchout

Our AASPatchout method is integrated into specific transformer blocks within the PaSST architecture. The workflow of the AASPatchout transformer block is illustrated in Fig. 1c. Initially, the model records the dimensions $F$ and $T$, representing the original 2D shape of the input sequence, and passes them into the block. Subsequently, within the AASPatchout transformer block, after applying the standard multi-head self-attention (MHSA), attention weights from the classification (CLS) token to all other tokens are extracted.

$$x_{\text{class}} = \text{Softmax}\left(\frac{q_{\text{class}} \cdot K^{\top}}{\sqrt{d}}\right) V = a \cdot V \qquad (1)$$

where $a$ represents the attention score vector from the CLS token's query vector $q_{\text{class}}$ to all other tokens in a single attention head. This score is derived from the key $(K)$ and value ($V$) matrices, and $d$ is the dimension of the query vector, while $x_{\text{class}}$ is the resulting output vector for the CLS token. Specifically, we compute the mean attention scores across H heads, generating an importance score for each patch.

$$\bar{a}_i = \frac{1}{H}\sum_{h=1}^{H} a_i^{(h)} \tag{2}$$

where $\bar{a}_i$ is the final importance score for the i-th patch, computed by averaging the single-head attention scores $a_i^{(h)}$ (the score for patch $i$ in head $h$) across the total number of attention heads $H$.

After obtaining the importance scores for all patches, the attention vector $\alpha_{\text{CLS}\rightarrow\text{patches}}$ is reshaped into an attention score map $M \in \mathbb{R}^{F\times T}$. Subsequently, the model computes the row importance scores $r_f$ and column importance scores $c_t$ by respectively averaging the attention scores in $M$ across the temporal and frequency dimensions.

$$r_f = \frac{1}{T}\sum_{t=1}^{T} \bar{a}_{f,t} \tag{3}$$

$$c_t = \frac{1}{F}\sum_{f=1}^{F} \bar{a}_{f,t} \tag{4}$$

Where $\bar{a}_{f,t}$ represents the average attention score for the patch at frequency row $f$ and time column $t$, while $F$ and $T$ are the total number of frequency rows and time columns in the patch grid.

These scores represent the relative importance of each frequency row and time column within the spectrogram, forming the basis for an importance metric guiding the Patchout process. Then, a minor Gaussian noise is added to both. Its purpose is not to significantly change the importance ranking. Instead, it introduces randomness into the patch selection with similar importance scores. This randomness acts as a regularizer to prevent overfitting from over-reliance on specific patches.

To utilize this metric, the patch sequence is reshaped back into its two-dimensional form ($F \times T$). Subsequently, only the intersection of the $top - m$ rows and $top - n$ columns is retained, while the remaining patches are dropped. This intersection of important rows and columns effectively preserves the regions identified as most critical by the attention mechanism at this layer. The retained patches are then flattened into a 1D sequence and input into subsequent layers. The dimensions of the patch grid are accordingly updated to $F' = m$ and $T' = n$ before being passed to the next transformer block. This structured retention strategy aligns closely with the two-dimensional structure of spectrograms, preserving the positional relationships among patches.

### C. Multi-stage Integration

Transformer layers at different depths focus on different types of information. Shallow layers attend more to global context, whereas deeper layers primarily concentrate on local features. Therefore, we extend the original one-time Patchout mechanism into a multi-stage process to fully leverage the hierarchical attention of the transformer. To achieve this, AASPatchout is integrated into multiple transformer blocks. By applying Patchout at various stages, the model selectively retains information most relevant to each hierarchical level. Simultaneously, this multi-stage approach prevents premature loss of information, thereby avoiding the limitation of the model's feature learning capacity.

## IV. Experiments and Results

### A. Evaluation Setup

We validated the proposed AASPatchout method for music classification on two representative datasets: OpenMIC2018 and GTZAN.

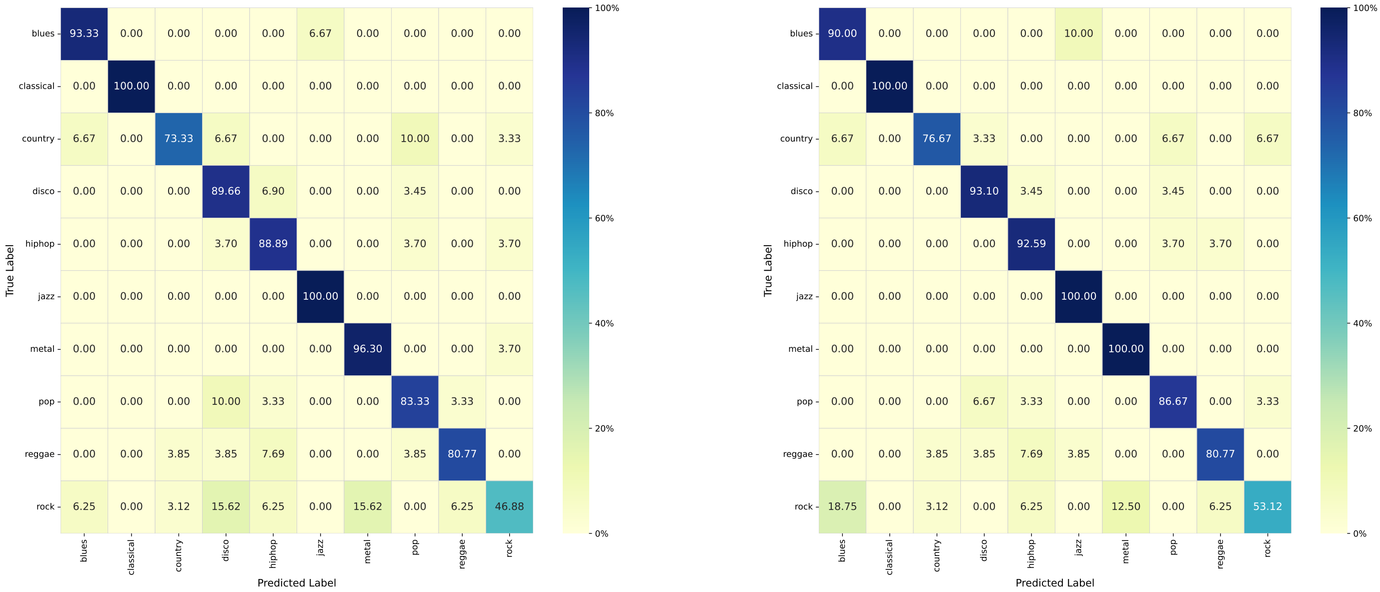
The OpenMIC-2018 dataset contains 20,000 10-second music excerpts sampled from the Free Music Archive (FMA) [19]. It is designed for multi-label instrument recognition [20], with each excerpt annotated for the presence or absence of 20 common instruments like piano, guitar, drums, etc.

The GTZAN dataset includes 1,000 30-second audio samples distributed equally across 10 genres. However, the GTZAN dataset has several inherent issues, for example, data duplication [21], [22]. Our experiments used a filtered and corrected dataset split. This split is now standard in GTZAN-based research to ensure a reliable evaluation of model generalization.

We evaluated the performance of our proposed AASPatchout method. For the polyphonic instrument recognition task on OpenMIC2018, we used mean average precision (mAP) and Area Under the ROC Curve (AUC) as the primary metrics. AUC assesses the model's ability to distinguish between classes, where its value closer to 1 indicates superior discriminative power. For the music genre classification task on GTZAN, we used accuracy as the evaluation metric. Additionally, we analyzed a confusion matrix to investigate inter-class misclassification patterns.

### B. Experimental Setup

We compared our proposed PaSST-AASPatchout model with PaSST-S (Structured Patchout), the best performing variant of the original PaSST baseline. To ensure the validity of experimental results, both models were trained using identical hyperparameters. Some of the experimental hyperparameters are set as follows, the input Mel-spectrogram sizes are $128 \times 998$ and $128 \times 3000$ respectively. Each spectrogram was divided into $16 \times 16$ patches, then linearly projected to 768 dimensions. The initial learning rate was fixed at 1e-5. Among them, in the most critical Patchout hyperparameter setting, for OpenMIC-2018, we dropped 3 frequency rows and 45 time columns, whereas for GTZAN we dropped 3 frequency rows and 120 time columns. Consequently, the patch dropout rate in both cases exceeds 50 %. Moreover, we applied Stochastic Weight Averaging (SWA) [23] during training to further improve model generalization. Furthermore, since AASPatchout introduces no new parameters or modules,



(a) Baseline

(b) Proposed Method

Fig. 2: Comparison plot of confusion matrices on the GTZAN dataset.The left figure (a) shows the baseline results and the right figure (b) shows the results of the proposed method.

re-pretraining on AudioSet was unnecessary. Consequently, both PaSST-AASPatchout and the baseline PaSST-S were initialized with the same pre-trained weights, and all layers were fine-tuned without being frozen.

### C. Results

TABLE I: Comparison on OpenMIC-2018 and GTZAN datasets.

| Model | OpenMIC-2018 | | GTZAN |
|---|---|---|---|
| | **mAP (%)** | **AUC (%)** | **Accuracy (%)** |
| PaSST-S | 84.30 | 90.88 | 84.84 |
| **PaSST-AASPatchout (proposed)** | **84.95** | **91.20** | **86.11** |

Table I show the comparison between the proposed PaSST-AASPatchout model and the baseline PaSST-S model on the OpenMIC-2018 and GTZAN datasets. PaSST-AASPatchout shows better performance than the baseline PaSST-S model on the primary evaluation metrics in both datasets. In detail, on OpenMIC-2018, PaSST-AASPatchout achieves an mAP of 84.95 %, exceeding the baseline by 0.65 %, and increases the AUC by 0.32 %. Similarly, on GTZAN, accuracy improved from 84.84 % to 86.11 %.

Fig. 2 shows the GTZAN confusion matrix obtained with baseline and PaSST-AASPatchout in one of the comparison experiments. The reduction in off-diagonal entries and the increase in diagonal accuracy indicate that the PaSST-AASPatchout model has a stronger ability to distinguish between easily confused genres. Specifically, the Rock category improved most, with accuracy rising from 47 % to 53 %. Initially, the baseline model misclassified Rock mainly as Disco and Metal, each at 16 %. In contrast, proposed model reduced Rock–Disco confusion from 16 % to 0 %. Although Rock–Blues confusion grew from 6 % to 19 %, this shift corrected several more severe errors. Spectrogram analysis reveals that Rock and Disco share regular drum-beat vertical stripes, low-frequency bass horizontal bands, mid-frequency vocal bands, and high-frequency cymbal wisps. However, Disco shows steady, whereas Rock features varied rhythms and larger dynamic contrasts. Thus, their overall structures align, but stripe spacing and intensity differ. Consequently, our model selects patches by attention importance, preserving these subtle cues better, learning Rock–Disco differences in drums and bass lines more effectively.

### D. Discussion

The above results showed that AASPatchout improved the performance compared to the baseline on both the OpenMIC-2018 and GTZAN datasets. This indicates that AASPatchout helps the model preserve longer continuous high-attention regions that might be disrupted by random Patchout. Moreover, the deterministic preservation of key feature regions suppresses genre ambiguity. It is no longer misled by superficial acoustic similarities, such as Rock and Disco beats; instead, it grasps deeper essential features. Although it faces new difficulties in separating some categories, it resolves several previously confusing classification issues, as observed in the confusion matrix. Furthermore, we conjecture that discarding low-attention patches reduces background interference, allowing the transformer to focus on salient melodic and rhythmic patterns. Overall, these outcomes demonstrate the efficacy of our AASPatchout strategy of preferentially retaining critical patches with high attention scores on music classification.

## V. Conclusions

Conventional Patchout randomly removes patches, limiting the PaSST's ability to capture key acoustic features. To address this, we proposed Attention-Adaptive Structured Patchout (AASPatchout). The method uses attention scores to decide which patches to keep or drop. It preferentially preserves critical acoustic features for music classification while adding no extra parameters or modules. Experimental validation on the OpenMIC2018 and GTZAN datasets confirmed that PaSST-AASPatchout achieved results superior to the baseline PaSST model. Therefore, this attention-based approach enhances the model's classification capabilities for music. These findings underscore the considerable potential of attention-guided, adaptive patch selection in improving performance on tasks involving music.

## References

[1] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions on speech and audio processing*, vol. 10, no. 7, pp. 504–516, 2002.

[2] M. Won, J. Spijkervet, and K. Choi, "Music classification: Beyond supervised learning, towards real-world applications," *arXiv preprint arXiv:2111.11636*, 2021.

[3] B. Ding, T. Zhang, C. Wang, G. Liu, J. Liang, R. Hu, Y. Wu, and D. Guo, "Acoustic scene classification: A comprehensive survey," *Expert Systems with Applications*, vol. 238, p. 121 902, 2024.

[4] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, p. 1, 2013.

[5] G. Peeters, S. McAdams, P. Herrera, G. Patella, D. Prazeres, A. M. Poulain, P. Susini, G. Lemaitre, N. Misdariis, F. Amadu, *et al.*, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," *CUIDADO Ist Project Report*, vol. 54, no. 0, pp. 1–25, 2004.

[6] M. Arsalan, "Transformers in Natural Language Processing: A Comprehensive Review," *International Journal for Research in Applied Science and Engineering Technology*, vol. 12, pp. 5591–5597, 2024.

[7] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[8] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Proc. Interspeech*, 2021, pp. 571–575.

[9] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection," in *Proc. ICASSP*, 2022, pp. 646–650.

[10] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," in *Proc. Interspeech*, 2022, pp. 2753–2757.

[11] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, "Ssast: Self-supervised audio spectrogram transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 10 699–10 709.

[12] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "BEATs: Audio Pre-Training with Acoustic Tokenizers," in *Proc. ICML*, vol. 202, 2023, pp. 5178–5193.

[13] S. R. Behera, A. Dhiman, K. Gowda, and A. S. Narayani, "FastAST: Accelerating Audio Spectrogram Transformer via Token Merging and Cross-Model Knowledge Distillation," in *Proc. Interspeech*, 2024, pp. 4733–4737.

[14] T. Lee and H. Lee, "Token Pruning in Audio Transformers: Optimizing Performance and Decoding Patch Importance," *arXiv preprint arXiv:2504.01690*, 2025.

[15] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie, "Not all patches are what you need: Expediting vision transformers via token reorganizations," in *Proc. ICLR*, 2022.

[16] E. Humphrey, S. Durand, and B. McFee, "OpenMIC-2018: An Open Data-set for Multiple Instrument Recognition," in *Proc. ISMIR*, 2018, pp. 438–444.

[17] G. Tzanetakis and P. R. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, 2002.

[18] J. B. Haurum, S. Escalera, G. W. Taylor, and T. B. Moeslund, "Which Tokens to Use? Investigating Token Reduction in Vision Transformers," in *Proc. ICCVW*, 2023, pp. 773–783.

[19] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A Dataset For Music Analysis," in *Proc. ISMIR*, 2017, pp. 316–323.

[20] D. Giannoulis, E. Benetos, A. Klapuri, and M. D. Plumbley, "Improving instrument recognition in polyphonic music through system integration," in *Proc. ICASSP*, 2014, pp. 5222–5226.

[21] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, 2012, pp. 7–12.

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

[23] P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson, "Averaging Weights Leads to Wider

Optima and Better Generalization," in *Proc. Uncertainty in Artificial Intelligence (UAI)*, 2018, pp. 876–885.