The first dataset describes the *"alberti bass"*: an accompaniment pattern first used during the classical period, where notes of chords are horizontally distributed in the left hand part [27]. For this dataset a semi-professional pianist composed ~25 second excerpts that contain this pattern, while trying to vary as much as possible other musical elements (e.g., key, tempo, content of the right hand).

The second concept is *"difficult-to-play music"*. A dataset of difficult musical excerpts was collected using the ranking produced by the musical score publisher G. Henle. [4] Excerpts were sampled from difficult pieces available in the MAESTRO dataset among different composers to avoid introducing biases toward some of them.

The third concept is *"contrapuntal texture"*, which denotes piano pieces composed of multiple monophonic voices that behave as separate instruments. This style is mostly present in pieces by some Baroque composers (e.g., Bach, Telemann, Händel, Buxtehude). For this dataset, we sampled Bach fugue performances, ensuring that they were not used during training the targeted composer classifier.

In addition to these three concept datasets, in this paper we use a collection of 10 different *random datasets*, which are built by randomly sampling 20 second excerpts from the MAESTRO dataset.

### 4.2 CAVs and Conceptual Sensitivity

A *Concept Activation Vector (CAV)* [12] $\mathbf{v}_l^k$ represents a concept $k$ in the output space of a neural network layer $l$. To compute it, we need the corresponding concept dataset (containing e.g., pieces with alberti bass) and a random dataset [12]. For a specific network layer $l$, we compute the layer activations (i.e., the output of the layer) for every piano roll $\mathbf{x}$ in the concept dataset, as well as the random dataset (see Figure 1). These activations can be seen as points in a $(H \times W \times C)$-dimensional space, where $H, W, C$ are the horizontal, vertical, and channel size in the layer activations tensor. We train a binary linear classifier (e.g., Support Vector Machine (SVM) or logistic regression) that separates the layer activations of the concept pieces from those of the random pieces. The vector of coefficients of this binary classifier, i.e., the vector orthogonal to the classification boundary, is the CAV $\mathbf{v}_l^k$ [12].

To measure whether a concept $k$ is relevant for a piece being classified as a certain composer $o$, we use the *conceptual sensitivity* $S_{k,o,l}$ of the system [12], i.e., the directional derivative of the prediction in the direction of the CAV,

$$S_{k,o,l} = \nabla g_{l,o}(f_l(\mathbf{x})) \cdot \mathbf{v}_l^k. \quad (1)$$

Here, $g_{l,o}$ transforms the activation vector $f_l(\mathbf{x})$ to the logit for the output class $o$, that is, it represents the remaining computations after a layer $l$ up to the output of the system. Intuitively, $S$ is a scalar that measures how much the output logits change if we perturb the layer activations in the direction of the CAV. Positive values mean that a concept $k$ encourages the classification of $\mathbf{x}$ as class $o$.

[4] https://www.henle.de/us/about-us/levels-of-difficulty-piano/

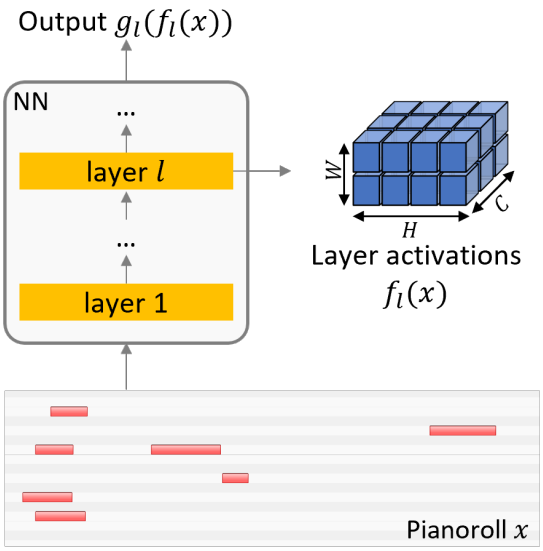


**Figure 1**. Layer $l$ activations for one piece excerpt. $H, W$ and $C$ are the horizontal, vertical, and channel size. The function encoded by the neural network is represented in two parts: $f_l$ from the NN input to the layer output, and $g_l$ from the layer output to the NN output.

The conceptual sensitivity is a local explanation, i.e., it explains how a system behaves for a specific input. We produce a global explanation, the TCAV score, that no longer depends on a specific input, by taking multiple pieces that belong to one class and computing the ratio of pieces for which $S$ is positive [12].

### 4.3 Experiments and Results

To investigate TCAV, we first compute a CAV for every one of our proposed concepts "alberti bass", "difficult-to-play music", and "contrapuntal texture". [5] As a linear classifier that separates the activations of the concept samples from those of the random samples, we use a linear SVM. This approach requires inputs of the same dimension, so we crop or pad all concepts to 20 seconds length (also used during training). Cropping is done by selecting the middle 20 seconds of a MIDI performance; padding adds silence until the appropriate length is reached.

In the next step, we examine the conceptual sensitivities of all the validation data and compute the TCAV score for all pieces by the same composer, i.e., the relative amount of positive conceptual sensitivities over the pieces. We again need to ensure inputs have the same length as our concepts, so we split every piece into non-overlapping 20 second segments and use all of these for subsequent computations. Although we can compute the TCAV score for any layer of the composer classifier, for brevity we show results of the penultimate layer subsequently, expecting this layer to encode the highest level features, similar to the image domain [28]. To compute TCAV scores, we perform ten runs with ten random datasets [12], and run a two-sided t-test and a Bonferroni correction for all concepts and composers

[5] Using https://captum.ai/api/concept.html

| | *Bach* | *Scarlatti* | *Haydn* | *Mozart* | *Beethoven* | *Schubert* | *Chopin* | *Schumann* | *Liszt* | *Brahms* | *Debussy* | *Scriabin* | *Rachmn.* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| alberti bass | + | | + | + | + | | - | | - | + | - | - | - |
| diff.-to-play music | - | | - | - | | | | + | + | | | + | + |
| contrapuntal texture | + | | | + | | - | - | | - | | - | | |

**Table 1**. Summary of TCAV scores for three concepts and the penultimate layer of a composer classifier. "+" indicate a positive influence of a concept on the classification of a composer, "-" negative influence. Empty cells show results that fail significance testing, i.e., the concept does not consistently en-/decourage the classification of a certain composer.

to validate our experiments. We use a significance threshold of $\alpha = 0.05/13$ (correcting for 13 hypothesis tests).

In our experiments, the SVM differentiating between concepts and random data has an accuracy greater than 0.9 for all concepts (in most cases, even 1). This means that the activations of the penultimate layer for the concept and the random samples are linearly separable, i.e., the CAV we produce represents the concept it is targeting. The TCAV score results are summarised in Table 1. All cells with "+" or "-" show results that pass our statistical significance test, and the remaining (empty) cells show results that fail. The symbol "+" means that a particular concept appears important for the classification of the corresponding composer (i.e., average TCAV score $> 0.5$); "-" that a concept discourages the classification of an input as a certain composer (i.e., average TCAV score $< 0.5$).

Table 1 shows both results that we would expect (e.g., alberti bass being relevant for Mozart, contrapuntal texture for Bach), and a few results which seem counter-intuitive (e.g., the model relying on alberti bass for Bach or Brahms – although one can find examples of such structures also in their works). It is possible that our model mixes the proposed concepts with other confounding ones, e.g., all pieces in the Alberti Bass dataset have also a quite simple harmonic structure. In general, there is no proof that the model *understands* our proposed concepts similarly to how a human listener would, yet we can make some interesting observations. Remarkably, even an extremely abstract concept such as "difficult-to-play music" might be grasped by our model: Liszt, Scriabin, Rachmaninoff are clear candidates for this attribute. The same applies to the "contrapuntal texture". Cases with negative impact ("-") should probably be interpreted with care: the fact that the classifier did not consider these concepts relevant for the classification does not necessarily mean that they are not present in the pieces. This could apply in particular to the subtle concept of contrapuntal texture. Also interesting is the case of Scarlatti, who is very much an outsider in classical music, style-wise ("a freakish if not downright incorrect composer" [29]) and could not be associated with any of our concepts.

## 5. UNSUPERVISED CONCEPT-BASED EXPLANATIONS

The supervised approach requires the user to pre-define concepts. This is very time-consuming, and the user could have to try a potentially infinite number of concepts if the network works differently than expected. In this section, we discuss an unsupervised approach instead: we build an explainer that identifies the relevant concepts and presents pieces where this concept is maximally activated (and some where the concept is not present). The musical expertise of the user is then used to translate these example pieces into a musical concept (with or without a name).

For the unsupervised approach described below, we introduce two limitations on the target neural network: we assume it to be convolutional and to use a non-negative activation function [30] (e.g., ReLU).

### 5.1 Tensor Factorisation for CAV Extraction

Consider a set of layer activations $\mathcal{X} = \{f_l(\mathbf{x}_1), \cdots, f_l(\mathbf{x}_N)\}$ generated from multiple pieces $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Layer activations that are close together (in terms of Euclidean distance) correspond to perceptually similar inputs [31] and therefore might describe similar concepts within the inputs. We could cluster similar activations and consider the pieces that generate these activations as examples of the same concept [32].

This works best if only one concept is present in a piece excerpt. However, we expect an excerpt to contain a number of different musical concepts, e.g., an alberti bass and a legato melody, both following a certain chord progression. Since these concepts can be shared across the same notes, we need a way to disentangle their effects on the layer activations. Due to the restriction on the type of activations (only non-negative) that we introduced in this section, we can use the NTD for this objective (see Section 5.3).

### 5.2 Channel CAVs

Given layer activations $f_l(\mathbf{x})$, let us consider their *channel-mode tubes* [33], i.e., the vectors obtained by fixing an index $h$ and $w$ for the horizontal and vertical dimension (see left-hand side of Figure 2). In the case of CNNs, we can consider each of these vectors as a different representation of the same piece with a different receptive field [34]. As proposed in [14], we can analyse channel-mode tubes $\in \mathbb{R}^C$ instead of full layer activations $\in \mathbb{R}^{H \times W \times C}$. This increases the

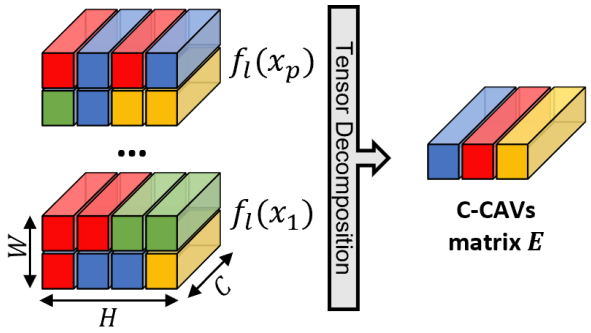


**Figure 2**. Left: Channel-mode tubes that result from fixing indices of activations in the $W$ and $H$ dimension of the activation space. Right: C-CAVs extraction from the layer activations of multiple pieces. Each channel tube is decomposed as a weighted sum of $C'$ C-CAVs.

amount of data and reduces the dimension of each data point by a factor of $H \times W$, therefore, we expect that the tensor decomposition that we will run on these data will achieve better results. Since every channel-mode tube represents a piece, we can compute CAVs in this restricted channel space $\mathbb{R}^C$, and refer to them as *Channel-CAVs* (or C-CAVs). We can again compute the conceptual sensitivities for such C-CAVs, as explained in Section 4.2.

We compute C-CAVs by starting from a dataset of pieces (segmented in 20-second excerpts) of the composers we want to explain (any number of composers can be considered). We input each excerpt into the trained system and produce activations for a certain layer $l$ (see Figure 1). The set of all activations can be seen as a tensor $\mathcal{X} \in \mathbb{R}^{N \times H \times W \times C}$, where $N$ is the number of piece excerpts and $H, W$ and $C$ are the frequency, time, and channel size of the layer activation tensor. We then apply a NTD to $\mathcal{X}$ to obtain a set of C-CAVs.

Moving to a channel-based formulation also permits us to highlight in which part of a piece a certain concept is present: since layer activations have a spatial correlation with input data [35] we can project the position $h, w$ of each C-CAV onto the input piano roll. This creates a *concept presence* heatmap showing the presence of a C-CAV [14] on a piano roll, which can be visualised to improve the user's understanding of the concept. Averaging the values in this heatmap gives a number that expresses how much a concept is activated in a certain excerpt and allows a ranking of the pieces according to their average concept presence.

### 5.3 Non-negative Tucker Decomposition

The NTD is a technique to decompose a tensor (e.g., $\mathcal{X}$) into a so-called non-negative *core* tensor $\mathcal{T}$, and multiple *factor* matrices $\mathbf{A} \in \mathbb{R}^{N \times N'}$, $\mathbf{B} \in \mathbb{R}^{H \times H'}$, $\mathbf{D} \in \mathbb{R}^{W \times W'}$ and $\mathbf{E} \in \mathbb{R}^{C \times C'}$ (one for every dimension) [33], such that

$$\mathcal{X} \approx \sum_{n=1}^{N'} \sum_{h=1}^{H'} \sum_{w=1}^{W'} \sum_{c=1}^{C'} t_{nhwc} \mathbf{a}_n \circ \mathbf{b}_h \circ \mathbf{d}_w \circ \mathbf{e}_c \quad (2)$$

Here, the core tensor $\mathcal{T}$ is in $\mathbb{R}^{N' \times H' \times W' \times C'}$, and one of its scalar elements is denoted by $t_{nhwc}$. The symbol "$\circ$" denotes the vector outer product of the column vectors of the (four) factor matrices $\mathbf{a}_n, \mathbf{b}_h, \mathbf{d}_w, \mathbf{e}_c$. The number of columns of the factor matrices (i.e., the NTD ranks), $N', H', W'$ and $C'$ are hyper-parameters that can be chosen by the user; if we set them to $N' = N$, $H' = H$, etc., an exact reproduction of the original tensor $\mathcal{X}$ is possible [33]. However, we mostly want to set them at lower values (i.e., $N' << N$, $H' << H$, etc.) to decrease the size of the matrices.

Equation 2 tells us that every channel-mode tube in $\mathcal{X}$ can be reconstructed as a weighted sum of the columns of matrix $\mathbf{E}$. As previously mentioned, each channel-mode tube represents a piece (in activation space), and each piece contains a sum of multiple concepts as C-CAVs. Then the C-CAVs we are looking for are disentangled as columns in $\mathbf{E}$ (see Figure 2), and their number is specified by rank C'.

The NTD also allows us to reconstruct an approximation of the original tensor (i.e., the original layer activations), and compute the output of the composer classifier by feeding it back into the network. We can then compute the ratio of the predictions that remain unchanged after the NTD step (i.e., the *fidelity*) [14] to evaluate its impact on the composer classifier. For more details on NTD, we refer to [33]; in this paper, we use the implementation provided in [36], with the Hierarchical non-negative Alternating Least Squares algorithm to update the factor matrices and the Fast Iterative Shrinkage-Thresholding Algorithm to update the core.

### 5.4 Experiments and Results

We compute the unsupervised explanations for the penultimate layer of our composer classification system and test multiple NTD ranks. As in [14], we present each concept to the user through the five piece excerpts with the highest average concept presence. The presentation of piece excerpts is more challenging for musical data than for images. Although symbolic performances can be visualised with piano rolls, some musical elements (e.g., harmonic elements) may be hard to understand in this format. We opt for a mixed audio-image visualisation where each excerpt is represented both with a piano roll (with a colour scale for velocity information) and with a listenable MIDI file. We create interactive piano roll visualisations (using Plotly [37]) in which the user can zoom in and out to explore different resolution levels. The concept presence heatmap is displayed as a semi-transparent mask over the piano roll. A heatmap with a fixed threshold, as proposed in [14], is hard to interpret for our data, so the user is presented with a slider that adapts the heatmap threshold (see Figure 3). We also provide "contrastive examples" for each concept, i.e., the 5 excerpts where the average concept presence is minimal. Although our explainer could find relevant concepts starting from a dataset that includes any number of composers, we focus on the results with only two composers. According to psychological studies [2], explanations are easier to understand when they involve only a small amount of information and when they target contrast cases [1], i.e., understanding why a composer is selected instead of another is easier than

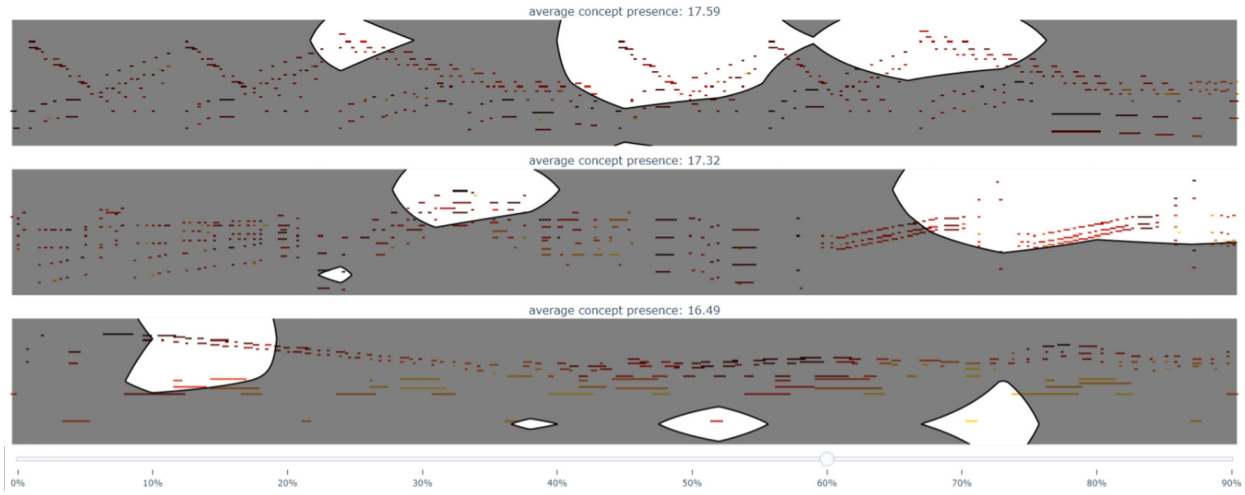


**Figure 3**. Visualisation of one concept (out of 4 produced by 4d NTD) through the masked piano rolls of the three dataset excerpts with the highest average concept presence. The concept heatmap threshold is set to 60%. This concept has positive conceptual sensitivity for Chopin and negative for Bach, therefore it is useful to distinguish between the two composers.

understanding why a composer is selected in general. For this reason, we also focus on C-CAVs with opposing conceptual sensitivities, i.e., negative for one class and positive for the other. We experimented with three different non-negative factorisation approaches: NTD applied to the 4d matrix (as explained in Section 5.3), NTD on a 3d matrix with concatenated horizontal and vertical dimensions, and NMF on a 2d matrix (as proposed by [14]) with concatenated horizontal, vertical, and piece dimension.

We found that the target classifier, when only two composers are considered, can be approximated with maximum fidelity by using only 3 to 5 C-CAVs, depending on the composers considered. For a fixed number of C-CAVs, we found no clear advantage for one of the three factorisation techniques with respect to the fidelity score. NTD allows for a much higher compression of $\mathcal{X}$, up to 15 times smaller, preserving the same fidelity; but it is also much slower to compute. From a manual analysis, we see that our unsupervised explainer finds, for each opposing concept, examples of typical composing styles that are useful for discriminating between two composers.

Figure 3 shows an example of what our model considers a typical Chopin-style pattern that is not present in Bach's music. In musical terms, it might be named "fast upward or downward movements (in the upper register) in parallel or broken thirds/sixths/octaves". Since our approach is based on non-negative factorisation techniques, some of their typical problems are also present in our results. For example, our system could produce one C-CAV that comprises what musical experts would typically interpret as two different concepts, or vice versa, produce two C-CAVs, both referring to the same concept. The former might have happened with the four small, seemingly unrelated blobs in the lower registers in the last two piano rolls of Figure 3. Furthermore, it is difficult to assign musically meaningful concept names to some C-CAVs, especially those with low average concept presence.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we explored a supervised and an unsupervised approach with the aim of producing explanations of deep musical classifiers interpretable by musicologists. In the supervised approach, we define high-level musical concepts (e.g., alberti bass) by building concept datasets and interrogate a classifier to find the relevance of a concept for the classifier decisions. This approach is useful when the user wants to test a specific concept. However, the process of creating a concept dataset can be time-consuming, requires high-level music expertise, and it could be necessary to try many different concepts before finding a relevant one. A solution to these problems is the unsupervised approach, which selects the relevant concepts by itself. Each concept is presented as a set of piece excerpts where the concept is maximally present. The user can listen to those excerpts and visualise them in a piano roll representation with a heatmap highlighting the concept position.

Future work on the supervised explainer will integrate recent promising results on model non-linearity and stricter hypothesis testing [38]. The unsupervised part will benefit from a formal user-based evaluation by musicologists to see which number of C-CAVs produce the most interpretable musical concepts and if there is agreement on their naming. Sparsity constraints applied to the core tensor and matrices in the NTD may attenuate the non-negative factorisation problems. While both supervised and unsupervised approaches work on piece excerpts of fixed length, an extension to variable length pieces could enable the study of concepts that span a longer time frame (e.g., piece structure). Moreover, our two approaches could be applied to explain audio classifiers, although this would complicate the visualisation of the concept heatmap for the unsupervised explainer, and could make the creation of concept datasets more challenging. Finally, dedicated user interfaces enabling to define concepts and visualise results would be helpful for musicologists.

## 7. ACKNOWLEDGEMENTS

This work was supported by the European Research Council (ERC) under the EU's Horizon 2020 research & innovation programme, grant agreement No. 101019375 (*Whither Music?*), the Austrian Science Fund (FWF, project No. P31988), and the Federal State of Upper Austria (LIT AI Lab).

## 8. REFERENCES

[1] T. Miller, "Explanation in Artificial Intelligence: Insights from the Social Sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[2] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[3] S. Mishra, B. L. Sturm, and S. Dixon, "Local Interpretable Model-agnostic Explanations for Music Content Analysis," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR*, 2017, pp. 537–543.

[4] S. Mishra, E. Benetos, B. L. Sturm, and S. Dixon, "Reliable Local Explanations for Machine Listening," in *Proceedings of the 2020 International Joint Conference on Neural Networks, IJCNN*. IEEE, 2020, pp. 1–8.

[5] V. Haunschmid, E. Manilow, and G. Widmer, "audioLIME: Listenable Explanations Using Source Separation," in *Proceedings of the 13th International Workshop on Machine Learning and Music, MML*, 2020, pp. 20–24.

[6] ——, "Towards Musically Meaningful Explanations Using Source Separation," *CoRR*, vol. abs/2009.02051, 2020.

[7] A. B. Melchiorre, V. Haunschmid, M. Schedl, and G. Widmer, "LEMONS: Listenable Explanations for Music recOmmeNder Systems," in *Advances in Information Retrieval: Proceedings of the 43rd European Conference on IR Research, ECIR*, vol. 12657. Springer, 2021, pp. 531–536.

[8] A. Ghorbani, A. Abid, and J. Y. Zou, "Interpretation of Neural Networks Is Fragile," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI*, 2019, pp. 3681–3688.

[9] P. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (Un)reliability of Saliency Methods," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, ser. Lecture Notes in Computer Science. Springer, 2019, vol. 11700, pp. 267–280.

[10] J. Adebayo, J. Gilmer, M. Muelly, I. J. Goodfellow, M. Hardt, and B. Kim, "Sanity Checks for Saliency Maps," in *Advances in Neural Information Processing Systems 31: Proceedings of the 32nd Annual Conference on Neural Information Processing Systems, NeurIPS*, 2018, pp. 9525–9536.

[11] V. Praher, K. Prinz, A. Flexer, and G. Widmer, "On the Veracity of Local, Model-agnostic Explanations in Audio Classification: Targeted Investigations with Adversarial Examples," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 531–538.

[12] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viégas, and R. Sayres, "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)," in *Proceedings of the 35th International Conference on Machine Learning, ICML*. PMLR, 2018, pp. 2673–2682.

[13] Z. Chen, Y. Bei, and C. Rudin, "Concept Whitening for Interpretable Image Recognition," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 772–782, 2020.

[14] R. Zhang, P. Madumal, T. Miller, K. A. Ehinger, and B. I. P. Rubinstein, "Invertible Concept-based Explanations for CNN Models with Non-negative Concept Activation Vectors," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI*, 2021, pp. 11 682–11 690.

[15] S. Kim, H. Lee, S. Park, J. Lee, and K. Choi, "Deep Composer Classification Using Symbolic Representation," *ISMIR Late Breaking and Demo Papers*, 2020.

[16] S. Chowdhury, A. Vall, V. Haunschmid, and G. Widmer, "Towards Explainable Music Emotion Recognition: The Route via Mid-level Features," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 237–243.

[17] P. Zinemanas, M. Rocamora, M. Miron, F. Font, and X. Serra, "An Interpretable Deep Learning Model for Automatic Sound Classification," *Electronics*, vol. 10, no. 7, p. 850, 2021.

[18] P. Zinemanas, M. Rocamora, E. Fonseca, F. Font, and X. Serra, "Toward Interpretable Polyphonic Sound Event Detection with Attention Maps Based on Local Prototypes," in *Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events, DCASE*, 2021, pp. 50–54.

[19] Z. Ren, T. T. Nguyen, and W. Nejdl, "Prototype Learning for Interpretable Respiratory Sound Analysis," in *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2022, pp. 9087–9091.

[20] A. R. Asokan, N. Kumar, A. V. Ragam, and S. S. Sharath, "Interpretability for Multimodal Emotion Recognition using Concept Activation Vectors," *CoRR*, vol. abs/2202.01072, 2022.

[21] J. Parekh, S. Parekh, P. Mozharovskyi, F. d'Alché-Buc, and G. Richard, "Listen to Interpret: Post-hoc Interpretability for Audio Networks with NMF," *CoRR*, vol. abs/2202.11479, 2022.

[22] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," in *Proceedings of the 7th International Conference on Learning Representations, ICLR*, 2019.

[23] Q. Kong, K. Choi, and Y. Wang, "Large-Scale MIDI-based Composer Classification," *CoRR*, vol. abs/2010.14805, 2020.

[24] D. Yang and T. Tsai, "Composer Classification With Cross-Modal Transfer Learning and Musically-Informed Augmentation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 802–809.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 2016, pp. 770–778.

[26] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," in *Proceedings of the 5th International Conference on Learning Representations, ICLR*, 2017.

[27] C. Rosen, *The Classical Style: Haydn, Mozart, Beethoven*. WW Norton & Company, 1997, no. 653.

[28] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Object Detectors Emerge in Deep Scene CNNs," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR*, 2015.

[29] R. Kirkpatrick, *Domenico Scarlatti: Revised Edition*. Princeton University Press, 1983, vol. 200.

[30] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *CoRR*, vol. abs/1811.03378, 2018.

[31] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 586–595.

[32] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, "Towards Automatic Concept-based Explanations," in *Advances in Neural Information Processing Systems 32: Proceedings of the 33rd Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019, pp. 9273–9282.

[33] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[34] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel, "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 29: Proceedings of the 30th Annual Conference on Neural Information Processing Systems, NIPS*, 2016, pp. 4898–4906.

[35] J. Dai, K. He, and J. Sun, "Convolutional Feature Masking for Joint Object and Stuff Segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 2015, pp. 3992–4000.

[36] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, "TensorLy: Tensor Learning in Python," *Journal of Machine Learning Research*, vol. 20, pp. 26:1–26:6, 2019.

[37] P. T. Inc. (2015) Collaborative Data Science. Montreal, QC. [Online]. Available: https://plot.ly

[38] J. Pfau, A. T. Young, J. Wei, M. L. Wei, and M. J. Keiser, "Robust Semantic Interpretability: Revisiting Concept Activation Vectors," *CoRR*, vol. abs/2104.02768, 2021.

# VERSE VERSUS CHORUS: STRUCTURE-AWARE FEATURE EXTRACTION FOR LYRICS-BASED GENRE RECOGNITION

**Maximilian Mayerl**[1] **Stefan Brandl**[2,3] **Günther Specht**[1]
**Markus Schedl**[2,3] **Eva Zangerle**[1]
[1] Department of Computer Science, Leopold-Franzens-Universität Innsbruck, Austria
[2] Institute of Computational Perception, Johannes Kepler Universität Linz, Austria
[3] Human-centered AI Group, AI Lab, Linz Institute of Technology (LIT), Austria
[1]{firstname.lastname}@uibk.ac.at [2,3]{firstname.lastname}@jku.at

## ABSTRACT

Lyrics-based genre recognition aims to automatically determine the genre of a given song based on its lyrics. Previous approaches for this task have commonly used textual features extracted from the entirety of a song's lyrics, neglecting the inherent structure of lyrics consisting of, for instance, verses and choruses. Therefore, we pose the hypothesis that features extracted from different parts of the lyrics can have significantly different predictive power. To test this hypothesis, we perform a series of experiments to determine whether models trained on features taken from verses and choruses perform differently for genre recognition. Our experiments indeed confirm our hypothesis, showing that generally, using features extracted from verses leads to higher performance than features extracted from choruses. Digging deeper, we found that this is especially true for *pop* and *rap* songs. *Rock* songs show the opposite effect, with features extracted from choruses performing better than those taken from verses.

## 1. INTRODUCTION AND RELATED WORK

In music information retrieval, genre recognition (also known as genre prediction or genre classification) is the task of automatically classifying the genre of a given song by assigning it to one or more predefined genres. This has a variety of applications, including the organization of music collections into easily recognizable categories, or using genre information as a feature in recommender systems. Over the years, many approaches have been developed for this task, most of which focus on audio-based genre recognition, i.e., using information extracted from the song's audio signal [1]. Less, but still substantial, work has also been done on lyrics-based approaches, which use features extracted from a song's lyrics (e.g. [2–5]). Lastly, hybrid approaches combining both audio and lyrics information have also been proposed (e.g., [6,7]), and it has been shown that audio and lyrics features are complementary and that using both together can lead to increased model performance. The approaches making use of lyrics information cover a wide range of feature types as well as underlying machine learning models.

For instance, Neumayer and Rauber [6] explored using lyrics features in conjunction with low-level audio features to detect the genre of songs. For the lyrics, they extracted feature vectors using the popular bag-of-words model and weighted the words using *tf-idf* weighting. They then performed classification via support vector machines. Mayer et al. [2] relied only on lyrics and extracted *tf-idf* weighted bag-of-words features, rhyme, and part-of-speech features as well as general statistical text descriptors for their approach. These features were then used to perform genre recognition using different machine learning models. Ying et al. [3] used information of the parts-of-speech used in a song's lyrics to recognize both genre and mood of a song. They used these features to train and evaluate three different machine learning models, namely support vector machines, k-nearest neighbor, and naive Bayes.

What these, and most other, lyrics-based approaches have in common is that they treat a song's lyrics as a uniform document, extracting features from the entirety of the song's lyrics. However, in reality, song lyrics have a structure. They consist of different parts, which can be divided into categories such as *intro*, *verse*, *chorus*, *bridge*, or *outro*. Leveraging such structural information, Tsaptsinos [4] proposed using a hierarchical neural network model using an attention mechanism. To the best of our knowledge, this is the only work that directly seeks to exploit song structure for genre recognition, making use of the hierarchical structure of song lyrics (words forming lines, lines forming segments, and segments, in turn, forming the song). Their results show that their model, which due to its attention mechanism can automatically learn on which parts of a song it should focus, outperforms existing approaches that extract and use features uniformly across the whole song. This shows that the location of features within a song plays an important role in genre recognition. However, their model applies attention at the word, line, and segment level only, and does not incorporate higher structural elements like *verse* or *chorus*. Fell

and Sporleder [5] used a variety of textual features for three distinct music classification tasks, including genre recognition. The features they employed include features describing information about a song's structure, like the number of repeated segments in a song and whether the song contains a chorus. Their results for genre recognition suggest that including such features can increase genre recognition performance, and therefore that song structure plays a role in determining a song's genre.

Inspired by those results, we formulate the following hypothesis: The structure of a song's lyrics plays a substantial role in genre recognition, and extracting features from different parts of the lyrics can lead to a significantly different performance of genre recognition models. To this end, we make the following contributions: (1) We construct a dataset of lyrics that contains the required information about lyrics' structure; (2) We perform a series of experiments, covering both feature sets as well as machine learning algorithms used in the past for genre recognition, and show that there indeed exists a significant difference in predictive performance between features extract from the verses of a song as opposed to the same features extracted from the choruses; and (3) We examine how that difference in performance depends on the concrete machine learning algorithm and feature set used, as well as on the genre.

The remainder of this paper is structured as follows. Section 2 explains how we created the dataset required for our experiments. In Section 3, we provide a detailed overview of our experiments. Following that, we discuss our results in Section 4 and finally provide a summary and outlook to future work in Section 5.

## 2. DATASET

To investigate the impact on classification performance of features from different structural elements of lyrics, we require a dataset that contains both lyrics data with structure information (i.e., information on which structural part of the song— verse, chorus, etc.—any particular line in the lyrics belong to) as well as genre tags. Since no such dataset is publicly available, we describe the creation of such a dataset in the following and subsequently, provide a statistical description of its contents.

### 2.1 Dataset Creation

We used the popular LFM-2b dataset [8] to obtain an initial list of songs. Using this list of songs, we then obtained lyrics data from *genius.com*. We chose this source because it not only provides lyrics for a large number of songs, but it also has an active community of users who annotate lyrics with structure information. In addition to lyrics, they also provide tags for many of the songs in their database. We use the primary tag as given by *genius.com* as our genre tag, and then filtered the list of songs such that only songs with one of the top five most occurring genre tags — *pop*, *rock*, *country*, *rap*, and *r&b* — remained. These steps provided us with an initial dataset of 2,135,504 songs with both genre and lyrics information.

| Property | Value |
|---|---|
| Number of songs | 295,416 |
| Number of artists | 39,357 |
| Number of tokens in all choruses | 193,696,032 |
| Number of tokens in all verses | 195,567,571 |
| Average number of choruses per song | 3.46 |
| Average number of verses per song | 2.37 |

**Table 1**: Summary statistics of our dataset.

We then performed a series of cleaning and transformation steps on this initial dataset. First, we expanded repeating parts of the lyrics that were given in short form; i.e., lyrics on *genius.com* frequently use annotations like *[x2]* to indicate that a given part (paragraph or line) in the lyrics should be repeated. We removed those repeating annotations and duplicated the corresponding parts in the lyrics text accordingly. Following that, we removed all other annotations that do not correspond to structural parts of the lyrics; for instance, instrument annotations or singer information. In the next step, we used `langdetect` version `1.0.9` as well as `polyglot` version `16.7.4` to remove all songs with non-English lyrics from the dataset. Finally, since the dataset at that point contained duplicates—i.e., songs with identical lyrics—we removed those. These duplicates exist because LFM-2b sometimes contains multiple versions of the same song, like covers by a different artist or versions recorded during live events. To decide which song among a given set of duplicates to keep, we used the number of listening events according to LFM-2b and kept the copy with the highest listening count. We chose this approach since the copy with the highest listening count is also most likely to have the most complete set of genre tags.

Following these cleaning and transformation steps, we split the lyrics of each song into its constituent structure parts. For this, we used the structure annotations provided by *genius.com*. These annotations specify which part of the song the subsequent text belongs to, with that part extending until the next structure annotation or until the end of the song. They often also give some additional information, like who sings the given part or which number the part has in the song. Examples of such annotations include *[Chorus]* or *[Verse 1: Austin Brown]*. For our splitting, we considered the following set of structural annotations: *verse*, *chorus*, *intro*, *outro*, *bridge*, *hook*, *refrain*, *interlude*, and *drop*. We also combined *chorus* and *refrain* and mapped both of those annotations to *chorus*. For our subsequent experiments, we considered only the *verse* and *chorus* parts of the songs in the dataset, but we retained the other parts for potential future work.

Subsequently, we removed all songs that did not consist of at least two parts. This gave us an intermediate dataset consisting of 416,945 songs. Finally, since our experiments focus on *verse* and *chorus*, we created a final dataset containing only those songs which have at least one verse and at least one chorus, yielding a total of 295,416 songs.