Designing Percussive Timbre Remappings: Negotiating Audio Representations and Evolving Parameter Spaces

Appendix

Anonymous Author(s)

A Audio Features

All features use frame-based processing and are computed in the frequency domain, except for loudness, which is computed in the time domain. Three different window sizes were considered: 256 samples, 4410 samples, 11025, and frame-based processing was conducted on each window of samples. FFT settings for all window sizes are summarized in Table 1. Each window is left and right padded with $p = f - h$ samples, where f is the frame size and h is the hop size, to ensure that each portion of the window is equally represented in the analysis. Each frame is windowed using a Hanning window. All FFT magnitudes are converted to power by squaring, except for Mel-bands, which defaults to use magnitude within FluCoMa. For loudness computation, the same frame and hop sizes as those listed in Table 1 were used.

Table 1: FFT Settings in Samples

Window Size	Frame Size	Hop Size	FFT Size
245	32	8	64
4410	64	16	128
11025	128	32	256

A.1 Hybrid Descriptors

Hybrid descriptors are an 8-dimensional feature. Pitch is computed using the YinFFT algorithm [3] with a minimum frequency of 40Hz and a maximum frequency of 10kHz. The output is in semitones and a pitch confidence value between 0.0 and 1.0 is generated for each frame. Two spectral features – spectral centroid and spectral flatness – are included, and are computed with a maximum frequency of 20kHz. Spectral centroid is in semitones and spectral flatness in decibels. Loudness is computing according to the EBU R128 specification [5], which involves first filtering with a k-weighting filter and then log-scaling.

Frame-based features are summarized using the following statistics. The median is used for pitch and outlier[1] frames are rejected using a ratio of 1.5. Spectral feature and loudness frames are summarized using the mean, and include the mean of the first order derivative. Pitch and loudness statistics are weighted on loudness.

[1]From FluCoMa documentation: "[Outliers are detected based on] a ratio of the inter quantile range (IQR) that defines a range from the median, outside of which data will be considered an outlier and not used to compute the statistical summary. For each frame, if a single value in any channel of that frame is considered an outlier (when compared to the rest of the values in its channel), the whole frame (on all channels) will not be used for statistical calculations."

NIME '25, June 24–27, 2025, Canberra, Australia

A.2 Mel-bands

Mel-bands are a 40-dimensional feature and include 40 Mel-bands. They are computed using a minimum frequency of 200Hz and a maximum frequency of 20kHz. The triangular Mel filters are normalized and the output is scaled to decibels. Mel-band frames are summarized using mean weighted on loudness.

A.3 MFCCs

MFCCs are a 104-dimensional feature. They include 13-coefficients computed on a 40-band Mel-band representation. The 13 coefficients start from the first coefficient, dropping the 0th which encodes signal amplitude. The minimum frequency is 200Hz and the maximum frequency is 12kHz. MFCC frames are summarized using the mean, standard deviation, minimum value, and maximum value. These statistics are also computed on the first order derivative. All statistics are weighted using loudness.

A.4 Spectral Shape

The spectral shape features are 14-dimensional. The spectral features included are the first four spectral moments (centroid, spread, skewness, and kurtosis), spectral flatness, spectral rolloff, and spectral crest. All spectral moments and spectral rolloff are computed using a log-frequency scale (semitones). Spectral flatness and spectral crest are output in decibels. Spectral rolloff represents the frequency under which n% of the signal energy is contained. For this work use a low spectral rolloff of 5%, which we found beneficial for timbre remapping, acting as a lower spectral anchor for sound transformations.

All spectral features frames are summarized over time using the mean and mean of the first order derivative, both weighted on loudness. We later discarded the first order derivative features based on the robust analysis, where these feature derivatives were deemed noisy with respect to our sound sources.

A.5 Perceptual Scaling

Wherever possible we have used feature scalings that are more perceptually relevant (e.g., decibel scales and log-frequency scales). A recent study has investigated scaling of different features and suggested scaling functions for a number of spectral features [4]. Three features are relevant: centroid, spread, and skewness. We empirically validated that spectral centroid in semitones is a reasonable fit to the suggested function in that work. We leave validation of perceptual scalings for the other features as future work.

B Hyperparameter Tuning

When training a regressor, even a relatively simple one like a multi-layer perceptron (MLP), one is faced with a plethora of design decisions related to hyperparameter selection. To mitigate this decision process we developed an external tool in Python to train MLPs and perform hyperparameter searches over the parameter space of FluCoMa's regressor. This Python tool uses datasets exported from FluCoMa in JSON format and performs

a hyperparameter search to find MLP parameters using a Tree-Structured Parzen Estimator (TPE) algorithm [2], a Bayesian optimization technique, implemented in the Optuna python package [1]. We matched FluCoMa's MLP architecture in PyTorch and performed hyperparameter searches over MLP parameters for each timescale regressor trained. Table 2 lists the hyperparameters that were optimized and the range of values selected from during the search. For a description of each hyperparamter please refer to the FluCoMa documention[2].

Input and output data is normalized prior to training. A validation set of 20% of the training dataset is withheld and network training is halted if the validation loss doesn't improve for 20 epochs, as is the case in FluCoMa when training with validation. Each regressor was trained for a maximum of 1000 epochs and hyperparameters we optimized over 100 trials. The best resulting trained neural network and normalizer parameters are saved, which can then be loaded and used in FluCoMa. This tool is open-source and available via our website[3]. Selected hyperparameters for each timescale regressor are shown in Tables 3, 4, 5, 6 for hybrid descriptors, Mel-bands, MFCCs, and spectral shape, respectively.

Table 2: Hyperparameters Optimized and Ranges

Parameter	Values
Number of Hidden Layers	$\{1, \ldots, 8\}$
Hidden Layer Size	$\{1, \ldots, 128\}$
Hidden Layer Activation	{sigmoid, ReLU, tanh}
Output Activation	{identity, sigmoid, ReLU, tanh}
Batch Size	{2, 4, 6, 8, 16, 32, 64}
Learning Rate	$[1e^{-5}, 1.0]$
Momentum	[0.0, 1.0]

Table 3: Hybrid Descriptor MLP Parameters

Parameter	256 ⟶ 4410	256 ⟶ 11025
Layers	{30, 50, 102, 105}	{106}
Hidden Activation	ReLU	tanh
Output Activation	tanh	tanh
Batch Size	4	32
Learning Rate	$1.2e^{-1}$	$1.1e^{-2}$
Momentum	0.41	0.81

Table 4: Mel-Band MLP Parameters

Parameter	256 ⟶ 4410	256 ⟶ 11025
Layers	{97, 99, 119, 38, 84}	{45, 58, 89, 75, 82, 10, 120}
Hidden Act.	ReLU	ReLU
Output Act.	sigmoid	tanh
Batch Size	16	4
Learning Rate	$4.0e^{-1}$	$1.7e^{-2}$
Momentum	0.66	0.35

[2]https://learn.flucoma.org/learn/mlp-parameters/
[3]To be released with paper

Table 5: MFCC MLP Parameters

Parameter	256 ⟶ 4410	256 ⟶ 11025
Layers	{113, 87}	{104}
Hidden Act.	ReLU	ReLU
Output Act.	sigmoid	sigmoid
Batch Size	2	2
Learning Rate	$2.5e^{-1}$	$1.9e^{-1}$
Momentum	0.26	0.81

Table 6: Spectral Shape MLP Parameters

Parameter	256 ⟶ 4410	256 ⟶ 11025
Layers	{85, 121}	{78, 127, 103, 27, 17}
Hidden Act.	ReLU	ReLU
Output Act.	sigmoid	ReLU
Batch Size	16	2
Learning Rate	$2.9e^{-1}$	$7.9e^{-3}$
Momentum	0.18	0.26

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*.

[2] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems*, Vol. 24. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html

[3] Paul Brossier. 2006. *Automatic annotation of musical audio for interactive applications.* Ph. D. Dissertation. Queen Mary University of London, London, UK.

[4] Savvas Kazazis, Philippe Depalle, and Stephen McAdams. 2022. Interval and Ratio Scaling of Spectral Audio Descriptors. *Frontiers in Psychology* 13 (March 2022). https://doi.org/10.3389/fpsyg.2022.835401

[5] International Telecommunication Union. 2006. Algorithms to measure audio programme loudness and true-peak audio level. *ITU-R BS.1770* (2006). https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-0-200607-S!!PDF-E.pdf