

Better music demixing with sliCQT

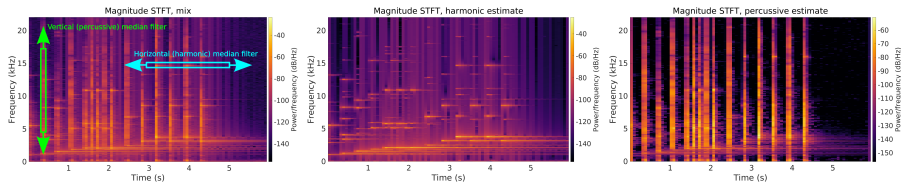
Submission to Cadenza Challenge CAD1

Sevag Hanssian

December 08, 2023

Time-frequency tradeoffs

Median-filtering harmonic/percussive source separation (HPSS)¹



- 1 Short window (256) for percussion, long window (4096) for harmonic
- 2 Short-time Fourier Transform (STFT) window size matters per-target² in VDBO problems
- 3 In musical and auditory contexts, frequency resolution should increase from high to low frequencies (vice-versa for time resolution)³
- 4 CQT⁴ uses long windows in low frequencies and short windows in high frequencies for the 12-tone Western pitch scale

1 Derry Fitzgerald. 2010; Jonathan Driedger et al. 2014.

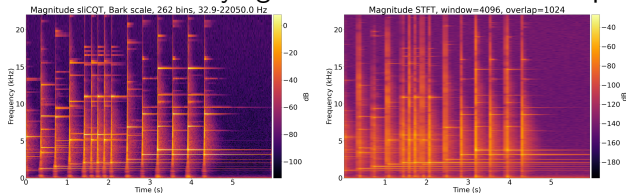
2 Ilya Kavalerov et al. IEEE, 2019.

3 Christian Schörkhuber et al. 2012; Monika Dörfler. PhD thesis. 2002.

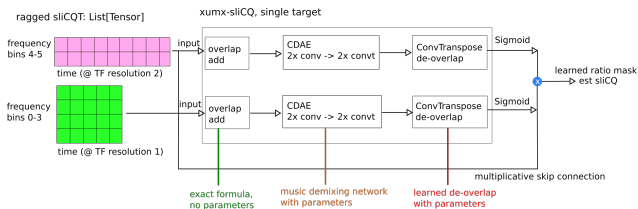
4 Judith Brown. (1991).

xumx-sliCQ v1 @ MDX 2021

- 1 sliCQ Transform⁵ is an STFT-like transform with **perfect inverse** that uses time-varying windows for nonuniform spectral analysis



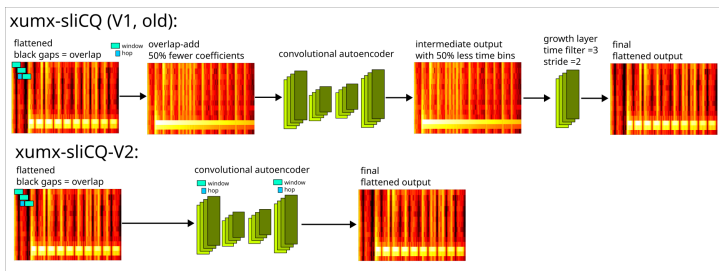
- 2 Bark scale (262 bins 32.9–22050 Hz) sliCQT + convolutional denoising autoencoder (CDAE) architecture⁶ to achieve 3.6 dB SDR



- 5 Gino Angelo Velasco et al. 2011; Nicki Holighaus et al. (2013).
- 6 Emad M. Grais et al. 2017.

xumx-sliCQ v2 @ CAD1 2023

- 1 Bark scale may have some benefits for human listeners
- 2 Focused solely on VDBO demixing problem
- 3 Better handling of overlap-add, mask sum loss, differentiable Wiener filtering, and complex MSE⁷: github.com/sevagh/xumx-sliCQ



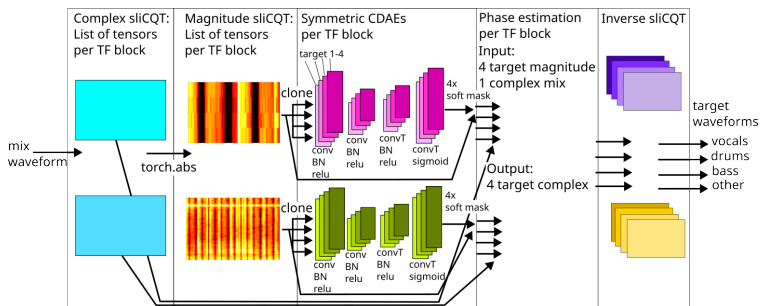
$$X_{\text{mix}} = X_v + X_d + X_b + X_o$$

$$|X|_{\text{mix}} = M_v |X|_{\text{mix}} + M_d |X|_{\text{mix}} + M_b |X|_{\text{mix}} + M_o |X|_{\text{mix}}$$

$$\rightarrow \mathbf{1} = M_v + M_d + M_b + M_o$$

⁷ Chin-Yun Yu et al. *arXiv preprint arXiv:2112.03752* (2021).

xumx-sliCQ v2: results



- 1 4.4 dB SDR up from 3.6
- 2 HAAQI score: mean of 0.094 vs. 0.255 of Baseline 1 (demucs)
- 3 BAQ score: mean of 41.84 vs. 41.40 of Baseline 1 (demucs)
- 4 More efficient (using bfloat16 for faster training and inference, etc.)
- 5 Weights are 60 MB
- 6 Fast realtime variant with 4.0 dB SDR using causal convolutions

New demixing-related project

Aim of these systems are to improve listening experience for those with different hearing.⁸ VDBO models are not very accessible (inscrutable Python errors, need >64GB RAM, etc.)

<https://freemusicdemixer.com>

Optimized C++ inference for UMX + Demucs, compiled to WebAssembly, running in the web, client-side on your browser, under 4 GB of memory

The screenshot shows the homepage of 'free-music-demixer'. At the top, it says 'free-music-demixer' and 'Split songs, demix music, and separate stems with our AI-based tool: free, private, and unlimited use directly in your browser'. Below this are navigation buttons for Home, About, Blog, Features, and a 'Request code' button. The main heading is 'Free AI-powered music demixer'. The text describes the tool's capabilities: 'In music demixing, music source separation, or song splitting, AI models are used to separate the different instruments from a music recording into stems. This web application allows you to demix or split your music files, free and with no usage limits since it runs on your computer!'. It also mentions 'Load a song to decompose it into bass, drums, vocals, melody, and karaoke using a next-size-of-the-art AI model. Open Source with the LMDXL pretrained weights. This site is created and maintained by Sevgi.' A diagram shows a 'Source Separator' box receiving an audio waveform and outputting five separate stems: drums, bass, vocals, melody, and karaoke. At the bottom, it says 'Runs locally in your browser!' and 'Unlike similar products, we're free to use and doesn't store your data. All processing is done in your browser, and your files are never uploaded anywhere to fully respect your privacy. It runs well on computers and very slowly on smartphones, user beware.'

The screenshot shows the 'BATCH DEMIX (Experimental)' interface. It prompts the user to 'Split all songs in a folder' and shows a 'Load file' button. Below that, it says 'demux: No directory selected' and has a 'Start batch demix' button. A progress bar indicates 'Batch demix progress...'. At the bottom, it says 'To correct the running js, refresh the page. BASH OKAY.' There is a 'Show dev logs' button. Below the interface is a terminal window showing the following output:

```
[ffmpegscript 27:00:40] web3 music demux is now ready
[ffmpegscript 27:00:40] beginning demux job
[ffmpegscript 28:12:00] Time: (class[1] demuxForm) | Layer 0
[ffmpegscript 28:12:07] Time: (class[1] demuxForm) | Layer 0
[ffmpegscript 28:12:07] Time: (class[1] demuxForm) | Layer 0
[ffmpegscript 28:12:07] (class[1] demuxForm) | File loaded
[ffmpegscript 28:12:08] Time: (class[1] demuxForm) |
[ffmpegscript 28:12:08] Time: (class[1] demuxForm) |
[ffmpegscript 28:12:08] Time: (class[1] demuxForm) |
[ffmpegscript 28:12:08] Time: (class[1] demuxForm) |
```

⁸<http://cadenzchallenge.org/about>