Perceptually Guided Music Enhancement for Hearing Aids Using Source Separation and Gain Optimization

Introduction and Background

Music listening can be especially challenging for individuals with hearing loss, as conventional hearing aids are often tuned for speech and may distort musical fidelity. The ICASSP 2024 Cadenza Challenge addresses this by focusing on *music enhancement for hearing aid users*. In the Cadenza scenario, a listener with hearing aids hears music played from stereo loudspeakers, and the goal is to personalize the music mix (e.g. boost vocals or other instruments) according to the listener's hearing profile. The challenge baseline system performs a *demix-and-remix* pipeline: it first separates the incoming stereo mixture into *vocals, drums, bass,* and *other* stems (using a pre-trained source separator), then applies *fixed gain* adjustments before remixing down to a stereo output. Finally, a standard hearing-aid frequency shaping (the NAL-R prescription computed from a person's audiogram) is applied for personalized amplification. This baseline approach is rule-based and not optimized for perceptual audio quality – it relies on generic gain settings and hearing aids speech-oriented filtering and amplification rather than learning from data.

A key metric for evaluating music enhancement in this context is the *Hearing-Aid Audio Quality Index (HAAQI)*. HAAQI is an intrusive objective metric that compares an enhanced signal to a "clean" reference, focusing on differences in the time-frequency envelope, temporal fine structure, and long-term spectrum [1]. Notably, HAAQI correlates better with hearing-impaired listeners' perceived quality than traditional fidelity metrics like signal-to-distortion ratio (SDR). In fact, improvements in conventional metrics do not always translate to higher HAAQI scores [2]. This project seeks to improve *perceptual quality* (as captured by HAAQI) beyond the baseline, by introducing learnable components and perceptual loss optimization in the Cadenza enhancement pipeline.

Proposed Approach: We build on the baseline system by integrating machine learning-based modules that can be trained end-to-end using a differentiable proxy for HAAQI. Our system replaces the baseline's fixed gain rules and pretrained source separator with a *personalized Gain Predictor network* and a *Source Separator* fine-tuned on the HRTF-processed input data, jointly optimized to maximize HAAQI. To enable this, we leverage **HAAQI-Net**, a recently proposed non-intrusive deep learning model that predicts HAAQI scores directly from audio and a listener's hearing profile. By using HAAQI-Net to estimate the perceptual quality of our system's output during training, we can back-propagate a *perceptual loss* signal – effectively training the model to "please" the HAAQI metric. We hypothesize that this perceptually guided learning will yield outputs that are qualitatively better for hearing-aid users, compared to optimizing conventional losses alone.

Datasets

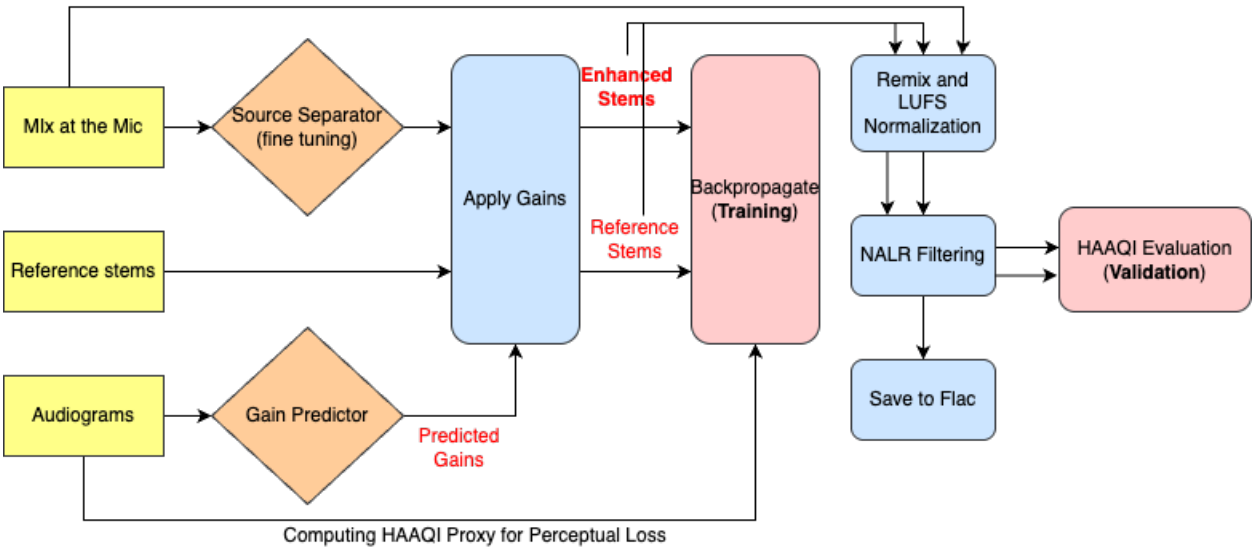
We used the official Cadenza Challenge datasets for training and evaluation. The **CAD_ICASSP_2024 Core** dataset served as our training set, and **CAD_ICASSP_2024 Validation** as the validation set. These datasets are derived from MUSDB18-HQ music tracks processed through hearing-aid acoustics. Each sample in the dataset consists of:

- **Binaural mixtures at the hearing aids** – two-channel audio simulating what the listener's hearing aid microphones pick up (including room effects and cross-talk between channels via head-related transfer functions).
- **Isolated clean stems (Vocal, Drums, Bass, Other)** for that same music piece, also rendered through the binaural hearing aid microphone simulation. These serve as *ground-truth reference stems* for training and evaluation.
- **Listener audiogram** – an 8-band audiogram profile for a particular listener, indicating hearing loss levels (in dB) at standard audiometric frequencies. In the training data, each "scene" pairs a music track (out of 100 songs each processed with 4 different HRTF simulations resulting 400 audio data) with two specific listener profile, enabling personalized enhancement, resulting 800 scenes in total.

The training set comprises a variety of songs and listener profiles, with both scene-listener-pair list and segment positions randomized in order, effectively providing diverse examples of training data within a small amount. We limited training to segments of 5 seconds from each track to manage memory and to augment the data (by sampling random segments per epoch). Each epoch trains 50 segments grouped in 10 batches of 5 segments. The validation set contains disjoint scenes (new combinations of music and listeners) for evaluating generalization. All audio is sampled at 44.1 kHz as per the challenge specification.

Methodology

System Architecture



Our enhancement system follows the same high-level pipeline as the baseline – *demix → modify gains → remix → amplify* – but with two critical learnable components:

- **Source Separator:** We use *Hybrid Demucs (HDemucs)* as our base separator, which is a state-of-the-art music demixing model operating in both time and frequency domains. Rather than using it as a fixed black box, we *fine-tuned* the separator on the task. To maintain the robust source decomposition ability of the pre-trained model while adapting to the new domain, we adopted a partial fine-tuning strategy: the Demucs encoder and mask estimation layers were frozen, and only the decoder (and final reconstruction layers) were updated during training. This allows the model to adjust its output slightly (e.g., to focus more on perceptually important aspects of each stem that maximizes HAAQI performance in a binaural setting) without unlearning the basic separation capability. The separator takes a stereo mixture and produces four output waveforms corresponding to vocals, drums, bass, and other instruments.
- **Personalized Gain Predictor:** Instead of applying fixed gains from a lookup table, we introduce a neural network to predict the optimal gain for each separated stem **given the listener's audiogram**. This Gain Predictor is a small convolutional neural network (CNN) that accepts the 8-band audiogram (for left and right ears) as input and outputs a set of gains (scalars) for the four source channels. These gains are applied multiplicatively to the corresponding Demucs output stems. Intuitively, this network learns how much to amplify each component of the music for a specific hearing loss pattern. For example, a user with high-frequency hearing loss might need higher gain on the "Other" stem (which often contains high-frequency instruments like guitars or synths), whereas a user who struggles with speech intelligibility might benefit from extra vocal gain. By learning from data, the Gain Predictor can potentially discover complex, non-linear gain strategies that outperform simple rules like NAL-R for music.

In our implementation, both **remixing** and the **NAL-R** stage were retained as post-processing operations applied after loss computation and gradient backpropagation. While these stages are non-differentiable due to reliance on NumPy and static signal processing, we decoupled them from the training graph by computing the loss directly on the separated and gain-scaled stems in tensor form. This design preserves the integrity of perceptual and signal-based losses while maintaining compatibility with the baseline's audio rendering for evaluation and listening.

Loss Functions and Training Objective

One of the main challenges was designing a loss that correlates with perceptual audio quality (HAAQI). We initially experimented with a multi-objective loss combining **waveform MSE**, **scale-invariant SDR**, and **loudness mismatch** – similar to losses used in source separation and loudness normalization tasks [3]. The MSE term encouraged the enhanced signal to be close to the reference signal (the ideal remix using ground-truth stems), SDR loss encouraged better source fidelity [4], and a loudness loss penalized any LUFS level difference between enhanced and reference (to enforce proper normalization). However, this conventional loss proved *misaligned with HAAQI*: we observed that while MSE and SDR could be improved, the resulting

HAAQI scores often *degraded*. In other words, blindly optimizing for waveform similarity or SDR did not guarantee perceptual quality improvements – confirming that HAAQI captures aspects of quality that are not covered by those traditional metrics. This is shown in Table 1, where decreasing MSE and SDR accompany a decreasing HAAQI.

training_scores

epoch	Total Loss	MSE	SDR	Loudness	HAAQI_left	HAAQ_right	HAAQI_avg	LUFS_diff
1	65.3371	37.8912	-146.9586	35.7668	0.5031	0.3715	0.4373	0.47
2	150.1612	62.0099	-109.0049	60.4239	0.4359	0.3563	0.3961	1.27
3	21.4533	25.4544	-178.8951	24.8337	0.3429	0.3454	0.3442	2.15
4	15.7957	15.7153	-96.2645	14.4799	0.3256	0.3337	0.3296	2.99
5	-17.1802	8.3965	-135.9322	7.6015	0.3253	0.3307	0.3280	2.98
6	-27.0492	3.5702	-122.2900		0.3247	0.3355	0.3301	2.75

Table 1 decreasing MSE and SDR and decreasing HAAQI

To directly optimize the perceptual metric, we incorporated **HAAQI-Net** [5] into the training loop. HAAQI-Net is a neural network that predicts the HAAQI score *given the enhanced audio and the listener's hearing profile*. Importantly, it does *not* require the reference signal as input – it produces a quality prediction based only on the processed signal (making it a non-intrusive model). We obtained a pre-trained HAAQI-Net model from the authors' implementation and integrated it as a differentiable loss module, perceptual loss. During training, after our system produces an enhanced output for a batch of data, we feed the output (in mono, as required by HAAQI-Net) and the corresponding audiogram into HAAQI-Net to get a *predicted HAAQI score*. The loss is defined as the negative of this score, i.e. **perceptual loss** = -HAAQI-pred. This loss term pushes the model to maximize the predicted HAAQI. We weighted this perceptual loss quite heavily in the total objective, reflecting its priority. The final *joint loss* used for training was a weighted sum:

Total loss = MSE + 0.3 *SDR_loss + 2.0 *loudness_loss + alpha * perceptual_loss

where alpha is the perceptual loss weight. In early experiments we tried alpha=1 and alpha=5, but ultimately alpha=10 gave the best results, strongly emphasizing perceptual quality. The other weights (0.3 and 2.0) were tuned so that the magnitudes of SDR and loudness loss contributions were on a similar scale as the MSE term during initial training epochs.

Optimization: Initially, we encountered gradient flow issues due to the non-differentiable operations inherited from the baseline pipeline. Specifically, operations like remixing, normalization, and NAL-R gain application were implemented using NumPy arrays and library calls that blocked backpropagation. To fix this, we rewrote key stages of the pipeline in PyTorch:

- The **Source Separator** was updated with a `forward` method accepting tensors and supporting fine-tuning of non-frozen layers.
- A new **Gain Predictor** model was introduced and fully integrated into the training loop.
- The **apply_gains** function was replaced with a fully differentiable version using PyTorch-only tensor operations, mapping predicted gains (in dB) to linear scale.

To preserve the original evaluation structure, we applied **NAL-R** and **remix** only as **post-processing** steps, not as part of the backpropagation graph. We found that the NAL-R step had minimal influence on training gradients, consistent with findings in the baseline.

Unlike earlier attempts using just MSE or loudness loss, the introduction of **HAAQI-Net** as a differentiable perceptual loss led to better alignment between training loss reduction and perceptual quality improvements. This non-intrusive model predicts HAAQI scores using BLSTM and BEATs features, eliminating the need for reference audio. We set the **perceptual loss weight α to 10**, based on preliminary tuning.

Throughout training, we logged several metrics: the MSE loss, SDR (with sign flipped, so higher SDR corresponds to lower loss), loudness difference, and the HAAQI-Net predicted score. We also ran the system on the validation set after each epoch to compute the true HAAQI using the official script (which compares the processed output to the ground-truth remix).

Evaluation and Results

Training the model with the perceptual loss led to clear improvements in the HAAQI metric on both training and validation data. **Figure 1** below shows the training curve of average HAAQI-Net prediction (perceptual score) versus epoch, along with the SDR metric, to illustrate their interaction.

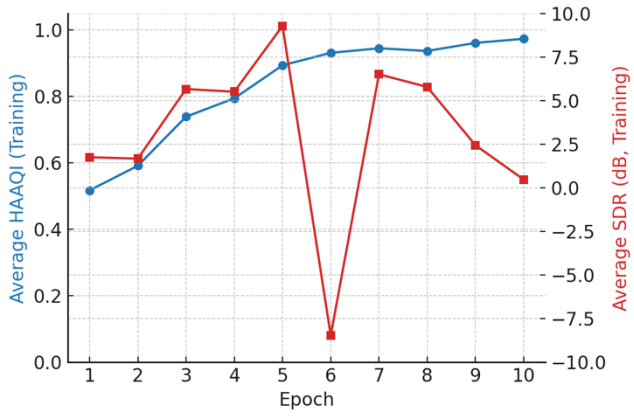


Figure 1: Training curves showing the model's perceptual quality (HAAQI score) and source fidelity (SDR) over 10 epochs.

HAAQI-Net pred (blue, left axis) steadily increases as the model learns to maximize perceptual quality, reaching approximately 0.97 by the final epoch. *SDR (red, right axis)* behaves less consistently – it improves initially but then fluctuates and even decreases in later epochs, indicating that the model sometimes sacrifices conventional source fidelity in favor of perceptual quality. Notably, at epoch 5 the SDR spiked (up to ~9 dB) then dropped sharply by epoch 6 (to around -8 dB), while HAAQI kept improving. This suggests that beyond a point, the model found a solution that sounds good to HAAQI-Net (high HAAQI score) but diverges significantly from the original source waveforms, thus lowering SDR. In subsequent epochs, SDR partially recovers but remains lower, whereas HAAQI continues to climb. This divergence highlights the fundamental tension between waveform fidelity and perceptual quality – our system prioritized the latter, as intended.

We monitored the true HAAQI on the validation set at each epoch to evaluate generalization. **Figure 2** shows the average left-ear and right-ear HAAQI scores on the validation set after each training epoch. Here we see a generally upward trend with training, though not as smooth as the training curve. The model's enhancements improved the validation HAAQI from an initial ~0.62 up to ~0.75 by epoch 10 (averaged across both ears).

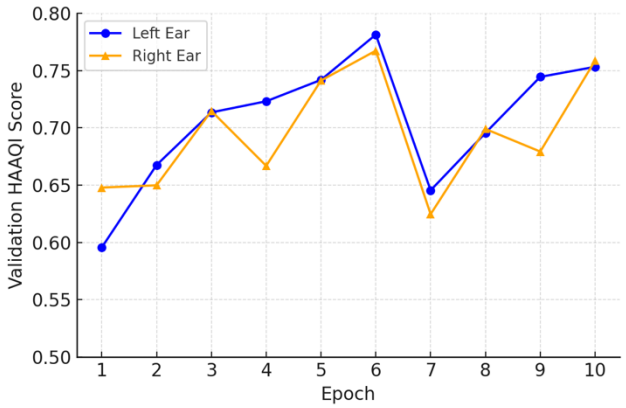


Figure 2: HAAQI scores on the validation set versus epoch, for left ear (blue) and right ear (orange).

Starting around 0.62 in epoch 1 (very close to the baseline system's performance), the score climbs to ~0.74–0.78 by epochs 5–6. There is a noticeable dip at epoch 7 (down to ~0.64), after which the scores recover and stabilize around the mid-0.7s. The drop in epoch 7 may indicate the model overfitting or a difficult set of validation examples; after adjusting (possibly benefiting from learning rate decay or simply the stochastic nature of training), the model rebounded in perceptual quality.

The final system (epoch 10) achieved an average HAAQI of **0.756**, significantly outperforming the challenge baseline (which was about 0.668 for the Demucs-based baseline [6]). This is a noteworthy gain in perceptual quality, given that a difference of 0.09 in HAAQI is quite substantial for this metric. For context, the baseline Open-Unmix system scored ~0.596 and the Demucs baseline ~0.668 overall – our system's score of ~0.756 represents about a 13% relative improvement over the best baseline. We emphasize that this improvement was achieved by optimizing directly for the perceptual metric, which appears to have guided the model towards more pleasing outputs (at least as far as HAAQI-Net and HAAQI can tell).

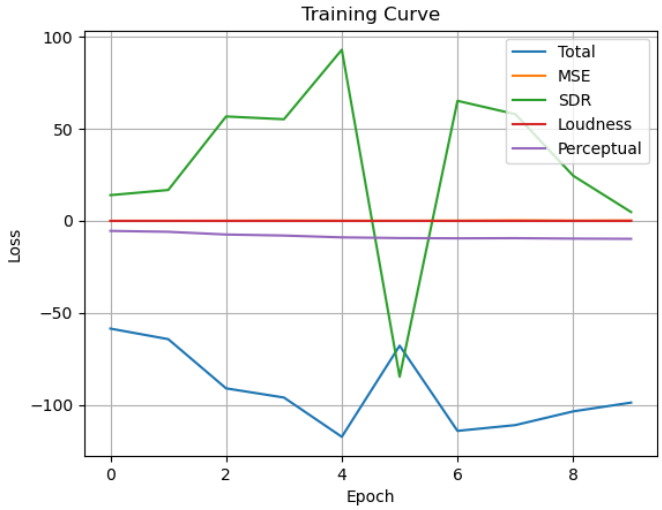


Figure 3: Training Losses (Total_loss, MSE_loss, SDR, Loudness_loss, Perceptual_loss)

Looking at other metrics, we found that the MSE between the enhanced and reference signals **increased** over training epochs, as seen in Table 2. This indicates the model did not simply learn to produce a waveform close to the reference. Instead, it likely introduced intentional deviations that HAAQI values positively – for example, emphasizing certain frequency bands or modulation patterns that improve perceptual clarity for the hearing-impaired listener, even if it means deviating from the exact reference waveform. The SDR metric on validation also did not monotonically improve; in fact, by epoch 10 the SDR for

some tracks was slightly worse than at epoch 1. However, our goal was perceptual quality, and indeed the HAAQI gains demonstrate success on that front.

Training Scores

epoch	total_loss	mse_loss	sdr_loss	loudness_loss	perceptual_loss	haaqi_score
1	-5.702146731890170	0.0023474228963515200	1.757067817908070	3.04740588395123E-05	-0.5177434774545530	0.5177434774545530
2	-6.417866277694700	0.0018917172448709200	1.6780812740325900	1.82284733782012E-05	-0.5916370093822480	0.5916370093822480
3	-9.089099884033210	0.0066328888526186	5.668739485740660	3.91941740872308E-05	-0.7395189404487610	0.7395189404487610
4	-9.586724328994750	0.01512801460921760	5.517599248886110	6.35100295710179E-05	-0.7946699380874630	0.7946699380874630
5	-11.71840238571170	0.01025554449297490	9.291289615631110	0.00018786731925498500	-0.8941646575927730	0.8941646575927730
6	-6.770863580703740	0.00865769654046741	-8.459224724769600	4.15950288811473E-05	-0.931737220287323	0.931737220287323
7	-11.39655885696410	0.01251984066299000	6.516709804534910	0.00039908386443126300	-0.9454863727092750	0.9454863727092750
8	-11.081882572174100	0.02968201912008230	5.793022346496580	0.00040518866426280 3	-0.9374468326568600	0.9374468326568600
9	-10.337636375427200	0.01446807290803 20	2.450789141654980	0.00014379690364874 70	-0.9617155134677890	0.9617155134677890
10	-9.861381530761740	0.02335900177713480	0.48061673641204700	9.84501646598624E-05	-0.9740752279758450	0.9740752279758450

Table 2: Training losses and HAAQI scores

Table 3 displays the validation scores. As it shows, loudness normalization was well maintained, with the LUFS difference between processed and target under 0.6 dB at all times (we allowed a small tolerance, and the loudness loss guided this). Informally listening to some examples, the enhanced music from our model sounded more vibrant and clearer (especially the vocals) compared to the baseline output, though occasionally some artifacts or over-enhancement of certain elements could be heard – a consequence of pushing the model to maximize a metric.

validation_scores

epoch	left_score	right_score	score	lufs_diff
1	0.5956419492108870	0.647893161733345	0.6217675554721160	-0.2848225349197730
2	0.6675961216434060	0.6498863898905600	0.6587412557669830	-0.5609164639404990
3	0.713512588325848	0.714739517653475	0.714126052989661	-0.164114643717682
4	0.723191033188158	0.666863148202838	0.695027090695498	-0.412950751447761
5	0.741886480400132	0.741315746768482	0.741601113584307	-0.0585931466396717
6	0.781355682301235	0.767337807520721	0.774346744910978	-0.0912237294246789
7	0.645496340271727	0.624692700828642	0.635094520550185	-0.161002572214187
8	0.695567968385229	0.699034964371448	0.697301466378339	-0.266304115127609
9	0.744496209532382	0.679311826091441	0.711904017811912	-0.216488710333325
10	0.753255207291361	0.758681450036215	0.755968328663788	-0.302715535622113

Table 3: Validation HAAQI scores and LUFS difference

Conclusion and Future Work

Due to time constraints, we did not run our final model on the held-out evaluation set. However, training and validation results suggest that using HAAQI-Net as a differentiable perceptual proxy leads to better alignment with perceptual quality goals than conventional loss functions alone.

By integrating a fine-tuned source separator with a learnable gain predictor, we developed a listener-personalized enhancement system that achieved higher HAAQI scores on the validation set compared to the static baseline. The results show that optimizing for perceptual metrics like HAAQI can lead to improved subjective quality, even if other metrics like MSE or SDR degrade.

HAAQI-Net proved to be a practical tool for perceptual optimization, but discrepancies (e.g., at epoch 7) suggest it is still an imperfect proxy. Future work could explore reinforcement learning or hybrid strategies incorporating true HAAQI scores as rewards to further close the gap.

On the personalization front, extending the input representation beyond audiograms—such as through learned listener embeddings—may better capture individual preferences. A differentiable dynamic range compressor could also be added to handle listener comfort in more flexible ways.

Ultimately, the effectiveness of any hearing aid system must be validated through user testing. Future work should include listening tests to ensure that objective gains translate into perceptual benefits and to gather feedback on artifacts or comfort issues that metrics like HAAQI-Net might overlook.

This project highlights how machine learning and perceptual modeling can reshape hearing aid audio processing, moving toward systems that are both personalized and perceptually optimized.

Bibliography

[1] "Hearing Aid Audio Quality Index (HAAQI) — Cadenza Tutorials." Accessed: Apr. 26, 2025. [Online]. Available: https://cadenzachallenge.org/cadenza_tutorials/metrics/haaqi.html

[2] K. H. Arehart, J. M. Kates, M. C. Anderson, and L. O. Harvey, "Effects of noise and distortion on speech quality judgments in normal-hearing and hearing-impaired listeners," *J. Acoust. Soc. Am.*, vol. 122, no. 2, pp. 1150–1164, Aug. 2007, doi: 10.1121/1.2754061.

[3] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music Source Separation in the Waveform Domain," Apr. 28, 2021, *arXiv*: arXiv:1911.13254. doi: 10.48550/arXiv.1911.13254.

[4] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – Half-baked or Well Done?," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 626–630. doi: 10.1109/ICASSP.2019.8683855.

[5] D. A. M. G. Wisnu, S. Rini, R. E. Zezario, H.-M. Wang, and Y. Tsao, "HAAQI-Net: A Non-intrusive Neural Music Audio Quality Assessment Model for Hearing Aids," Jan. 09, 2025, *arXiv*: arXiv:2401.01145. doi: 10.48550/arXiv.2401.01145.

[6] "The ICASSP SP Cadenza Challenge: Music Demixing/Remixing for Hearing Aids." Accessed: May 19, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10626340