Proceedings of the 29th International Conference on Digital Audio Effects (DAFx26), Cambridge, MA, USA, 1–4 September 2026

GRADIENT DESCENT OPTIMIZATION OF ROOM IMPULSE RESPONSES WITH PARAMETER-EFFICIENT DIFFERENTIABLE FEEDBACK DELAY NETWORKS

Ilias Ibnyahya and Joshua D. Reiss

Centre for Digital Music
Queen Mary University of London
London, UK
i.ibnyahya@qmul.ac.uk | joshua.reiss@qmul.ac.uk

ABSTRACT

Artificial reverberation can be produced either by convolving a signal with a measured room impulse response (RIR) or by synthesizing it with a parametric algorithm such as a Feedback Delay Network (FDN). The former reproduces a captured space faithfully but is costly to run and offers no control over its acoustic properties, while the latter is efficient and editable but hard to match to a specific room. In this paper we bridge the two by fitting a fully differentiable FDN to a measured RIR through gradient descent. The proposed network uses sixteen delay lines at a sampling rate of 48 kHz and trains all of its components jointly, including the delay lengths, the feedback matrix, the early-reflection taps, and a set of attenuation filters that control the frequency-dependent decay. The attenuation filters are realized as a single prototype of parametric equalizer bands shared across all delay lines, with per-line gains fixed by a proportionality condition. This design decouples reverberation time accuracy from the filter structure for a matched parameter budget, reaches a lower reverberation time error than graphic equalizer filters while requiring fewer multiplications per output sample and training several times faster. The network is optimized with a composite loss covering the temporal and spectral characteristics of the target. We evaluate the method on nine measured RIRs ranging from a small studio to a cathedral, and compare it against three baselines: another differentiable FDN, an analysis-synthesis pipeline, and a noise-shaped reverberator. The proposed system attains the lowest error on standard room-acoustic measures, the direct-to-reverberant ratio, and a multi-resolution spectral distance, and an ablation confirms the contribution of each trainable component. The resulting reverberator is fully parametric, uses comparable computation per sample to partitioned convolution at roughly 11 times less memory, and trains in under two minutes per RIR on a single consumer GPU.

1. INTRODUCTION

Realistic reverberation is central to immersive audio, music production, and virtual acoustics. Two approaches are typically used to apply a reverberation effect to an audio signal. Sample-based methods convolve a measured Room Impulse Response (RIR) with the input signal, whereas parametric methods synthesize reverberation with delay lines and feedback loops that build up echo density. Sample-based methods reproduce a captured space accurately but offer limited control over its properties and expose no parameters. Parametric methods are compact and editable in real time, but are less faithful to a specific target room. In this paper, we bridge the two by fitting a parametric reverberator to a measured RIR through gradient descent optimization.

A measured RIR, obtained by deconvolving an excitation such as an exponential sine sweep or maximum-length sequence [1], has two practical limitations. First, it captures a linear time-invariant response at a single source-listener position. Second, a measured RIR provides no interpretable parameters, so an engineer cannot modify the acoustic properties of the space for audio production or spatially-varying applications. A solution would be to store a large dataset of RIRs covering a range of positions or conditions, but this would still offer no parametric control. Real-time convolution with RIRs, typically partitioned block convolution [2], remains demanding for multichannel, spatially-varying, and hardware-constrained systems. Both limitations are mitigated by algorithmic reverberators such as Feedback Delay Networks (FDNs) [3], which model reverberation with a network of delay lines coupled by a feedback matrix. FDNs are an efficient and scalable structure for late reverberation, but they need their internal coefficients and parameters to be set to achieve good sounding reverberation. This is a non-trivial task, and the resulting reverberation may not match a realistic target.

FDN parameters were first set by analytical and heuristic rules [4] which led to automated matching methods, such as gradient-free genetic algorithms [5]. Dal Santo et al. introduced the RIR2FDN pipeline [6] which optimize a target RIR for low coloration, accurate reverberation time control and maximal echo density, combining state-of-the-art methods [7, 8] into an unified analysis-synthesis workflow using the FLAMO framework [9]. These analysis-synthesis methods scale poorly to high-order delay networks and do not provide an end-to-end pipeline suitable for gradient-based methods such as neural networks.

Differentiable Digital Signal Processing [10] enabled gradient-based tuning, but feedback systems are recursive and not directly differentiable. Lee et al. [11] resolved this with the frequency-sampling method [12], building the first end-to-end differentiable FDN. That model while innovative, is limited to six fixed length delay lines, shares a single attenuation filter response across them, giving limited interpretable reverberation time and echo density control. Mezza et al. made the delay lengths trainable [13] and later the FIR attenuation filters [14, 15], but evaluate only on a few RIRs at low sampling rates, on small FDNs, without parametric reverberation time control. Jot [16] introduced proportional attenuation filters for accurate reverberation time control. These are now usually realized as one graphic equalizer (GEQ) per delay line, with a two-stage shelving filter to correct

Copyright: © 2026 Ilias Ibnyahya et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, adaptation, and reproduction in any medium, provided the original author and source are credited.

the spectrum edges [8], or as differentiable parametric equalizers (PEQs) [17], which we review in Section 2.2. All of these methods focus on matching the late-reverberation tail and lack an early-reflection component.

In this paper we scale this paradigm to a sixteen delay-line, end-to-end differentiable FDN built around a proportional PEQ attenuation filter (Fig. 1) at a sampling rate of 48 kHz. A single prototype set of PEQ bands is shared across all delay lines, with per-line gains fixed by the proportionality condition [16]. This reduces the number of trainable parameters and bands per delay line, shrinking the search space and decoupling reverberation-time accuracy from the filter architecture.

Our contributions are threefold. First, we present a fully differentiable FDN architecture suitable for end-to-end gradient-based optimization, in which the delay-line lengths, feedback matrix, and early-reflection taps are all trainable, together with a scalable proportional PEQ attenuation filter for reverberation-time control. The recommended proportional PEQ configuration reaches a lower reverberation-time error than matched-parameter GEQ filters while using $2.8\times$ fewer multiplications per output sample and training faster. Second, we build a composite RIR-matching loss centered on an energy decay relief term. A controlled comparison shows that this loss improves even the noise-shaped baseline, which separates the contribution of the loss from that of the architecture. Third, we evaluate the system against three baselines on ISO 3382-1 metrics, a multi-resolution STFT distance, and a joint time-frequency EDR error, and an ablation over the early-reflection branch, the optimized delay lengths, and the feedback matrix confirms the importance of each element for accurate RIR matching. The source code, audio examples, and RIRs are available at `github.com/ilias-audio/DAFX26_DiffReverb`.

2. BACKGROUND

2.1. Feedback Delay Networks

A single-input, single-output (SISO) feedback delay network consists of N parallel delay lines of lengths $m_1, \ldots, m_N$ samples, coupled through a feedback path. Its transfer function is

$$H(z) = \boldsymbol{c}^\top \big[\boldsymbol{D}_m(z)^{-1} - \boldsymbol{U}\,\boldsymbol{\Gamma}(z)\big]^{-1}\boldsymbol{b} + d, \tag{1}$$

where $\boldsymbol{D}_m(z) = \mathrm{diag}(z^{-m_1}, \ldots, z^{-m_N})$ is the delay matrix, $\boldsymbol{U}$ is an $N \times N$ unitary feedback (mixing) matrix, $\boldsymbol{\Gamma}(z) = \mathrm{diag}\big(\Gamma_1(z), \ldots, \Gamma_N(z)\big)$ collects the per-delay-line attenuation filters, $\boldsymbol{b}$ and $\boldsymbol{c}$ are input and output gain vectors, and d is the direct-path gain. The feedback matrix $\boldsymbol{U}$ controls coupling between delay lines and affects echo density and coloration [7]. By default $\boldsymbol{U}$ is a normalized Hadamard matrix of order N, which guarantees losslessness regardless of the delay lengths.

2.2. Attenuation and Tone Filters

For the FDN to have a well-defined frequency-dependent reverberation time T_{30} regardless of which internal path a signal component follows, the attenuation filters must satisfy the proportionality condition [3]. On the unit circle $z = e^{j\omega}$ the condition is

$$|\Gamma_n(z)| = |\Gamma_{\text{prot}}(z)|^{\,m_n}, \tag{2}$$

where $\Gamma_{\text{prot}}(z)$ is a prototype per-sample attenuation filter shared across all delay lines and m_n is the length of delay line n in samples. In decibels the prototype gain at frequency bin k is

$$\gamma_{\text{dB}}(k) = \frac{-60}{f_{\text{s}}\, T_{30}(k)}, \tag{3}$$

so the attenuation of delay line n becomes $G_{n,\text{dB}}(k) = m_n \cdot \gamma_{\text{dB}}(k)$. This linear scaling in the log-magnitude domain is the property we exploit in our scalable PEQ parametrization in Section 3.3.

The prototype attenuation Γ_{prot} can be realized in several ways. The standard choice is a GEQ, a bank of band filters at fixed center frequencies and quality factors whose gains alone are tuned to the target decay [18]. One GEQ is placed on each delay line, so a B-band GEQ on an N-line FDN has $B \times N$ trainable gains. For a 16-line octave-band configuration this is $10 \times 16 = 160$ bands, rising to $30 \times 16 = 480$ at third-octave resolution. GEQ designs also lose accuracy at the low and high edges of the spectrum. A two-stage attenuation filter addresses this by cascading a low-order shelving pre-filter, which sets the broad spectral tilt, with a graphic equalizer that fine-tunes the remaining bands [8].

A PEQ instead exposes the center frequency, quality factor, and gain of every band as free parameters [17, 16]. Because the bands can move, a PEQ matches a target response with fewer sections than a fixed-grid GEQ, and a single prototype set of bands can be shared across all delay lines, with only the per-line gains following the proportionality condition (Eq. 2). We adopt this realization (Section 3.3) and ablate it against a GEQ in the same FDN in Section 5.4.

A tone filter is a separate equalizer placed outside the feedback loop. It corrects the residual spectral coloration of the output without changing the decay rate [16]. We use a trainable PEQ tone filter, described in Section 3.3.

3. PROPOSED METHOD

3.1. Differentiable FDN Architecture

Recursive IIR structures are hard to train by backpropagation [19], since each output feeds back as an input. The Frequency-Sampling Method [12], used in Lee et al. [11] and provided by the FLAMO framework [9], converts the recursive system into an FIR approximation of length N_{FFT}, so we backpropagate without truncated backpropagation through time. We also apply an anti-aliasing decay envelope in the time domain to suppress energy remaining at the N_{FFT} window boundary [9].

As shown in Figure 1, the proposed architecture comprises two branches. Branch A handles the late reverberation, with an N-element input gain vector $\boldsymbol{b}$, a recursive core with N integer delay lines coupled by a feedback matrix $\boldsymbol{U}$ and the scalable PEQ attenuation filters $\Gamma_n(z)$, an N-element output gain vector $\boldsymbol{c}$, and a post-loop tone-shaping PEQ filter $T(z)$. The mixing matrix and the delay lengths are both differentiable and can be held fixed or trained jointly with the filters. The fixed configuration sets the mixing matrix to a normalized Hadamard matrix and the delays to co-prime integers. Our configuration instead trains an orthogonal mixing matrix, parameterized as the matrix exponential of a skew-symmetric matrix to preserve losslessness, together with the delay lengths, in a single-stage optimization that fits the matrix and the filters together, unlike the scattering feedback matrices of

RIR2FDN [6]. Branch B models the direct sound and early reflections as a trainable sparse tap-delay line (Section 3.2) and can be disabled. The two branches are summed and passed through a trainable bandpass.

These choices define the configurations compared in Section 5.4. *Proposed (all-diff)* trains every component, namely the delay lengths, the mixing matrix, the attenuation filters, and the early-reflection taps. *FDN (all-diff, no ER)* is the same but with Branch B disabled. *FDN (fixed delay, train mix)* freezes the delays at their co-prime integers but trains the orthogonal mixing matrix, keeping the ER taps. *FDN (fixed)* freezes both the Hadamard matrix and the delays, training only the filters, with fixed ER taps; *FDN (fixed, no ER)* additionally disables Branch B. The filters (attenuation and tone), the input gains $\boldsymbol{b}$, the output gains $\boldsymbol{c}$ and direct path $\boldsymbol{d}$ are trained in every configuration.

3.2. Early-Reflection Taps

Branch B models the direct sound and early reflections with a sparse tapped-delay line, a classic early-reflection structure in artificial reverberation [20]. We use $P = 64$ taps with a maximal delay of 50 ms. Each tap p has a trainable gain g_p and a trainable delay τ_p. The branch response, realized in the frequency-sampling pipeline, is

$$H_{\mathrm{B}}(z) = \sum_{p=1}^{P} g_p \, z^{-\tau_p}. \tag{4}$$

The tap positions are initialized at the largest peaks of the target RIR. This sparse parameterization models discrete early reflections with $2P$ trainable values.

3.3. Attenuation and Tone Filters

We realize the prototype attenuation as a trainable PEQ (Section 2.2) defined by B bands, each with three trainable parameters stored as unconstrained reals $\theta_i = (\theta_i^f,\ \theta_i^q,\ \theta_i^\gamma)$ for $i = 1, \ldots, B$. These are mapped to physical quantities via a sigmoid function $\sigma(\cdot)$ followed by denormalization

$$f_i = \exp\big(\log f_{\min} + \sigma(\theta_i^f)\big[\log f_{\max} - \log f_{\min}\big]\big), \tag{5}$$

$$Q_i = \exp\big(\log Q_{\min} + \sigma(\theta_i^q)\big[\log Q_{\max} - \log Q_{\min}\big]\big), \tag{6}$$

$$\gamma_i = \gamma_{\min} + \sigma(\theta_i^\gamma)\big(\gamma_{\max} - \gamma_{\min}\big). \tag{7}$$

For shelf bands, Q_i in (6) is replaced by a shelf-slope parameter $S_i = S_{\min} + \sigma(\theta_i^q)(S_{\max} - S_{\min})$ on a linear scale. The numeric bounds are $f_{\min} = 20$ Hz and $f_{\max} = 20$ kHz, with $Q_{\min} = 0.2$ and $Q_{\max} = 2.0$ for peaking bands, $S_{\min} = 0.1$ and $S_{\max} = 0.95$ for shelves, and $\gamma_{\min} = 0.05$ s and $\gamma_{\max} = 100$ s, where γ_i represents the target reverberation time of band i.

The proto-reverberation-time γ_i is converted to a per-sample attenuation rate. The per-band gain applied to delay line n of length m_n samples is

$$G_{i,n} = \frac{-60}{f_{\mathrm{s}} \cdot \gamma_i} \cdot m_n, \tag{8}$$

the proportionality condition (2) applied per band. Gains are clamped to $[-30,\ 0)$ dB for loop stability. Because the frequencies and quality factors are shared across all delay lines, the attenuation stage has $B \times 3$ trainable parameters, independent of the FDN order N.

The B biquad sections are arranged as one low shelf, $B-2$ peak filters, and one high shelf, following the RBJ cookbook [21]. Band frequencies are initialized to a log-spaced grid from 100 Hz to 10 kHz for the peaking bands, with the low shelf at 100 Hz and the high shelf at 12 kHz. Proto-gains are initialized so that $\sigma(\theta_i^\gamma)$ maps to an T_{30} of 3 s, placing the initial operating point in a stable region with good gradient flow. The proposed PEQ parametrization is not limited to the RBJ biquad topology. Alternative filter structures such as state-variable filters may offer practical advantages in fixed-point or low-precision implementations, and the number and type of shelf and peak filters is variable.

The tone filter $T(z)$ is the same PEQ cascade ($B = 10$ by default) placed post-loop, inside Branch A after the output gains but before the branch sum and bandpass (Figure 1), so the attenuation filters alone control the frequency-dependent decay while $T(z)$ corrects the residual output coloration. Its gains are clamped to $[-30, +18]$ dB, and the output gains are scaled by $1/\sqrt{N}$ to normalize the energy extracted from the loop.

A trainable bandpass is applied after summing the branches, formed by a high-pass and a low-pass stage, each a 4th-order Butterworth IIR filter. Only the cutoffs f_{lo} and f_{hi} (in log-space, initialized to 100 Hz and 18 kHz) are trainable. Applying it to the summed branches shapes the direct path consistently with the late reverberation and suppresses DC and Nyquist-region artifacts if the target RIR has very low energy at those frequencies.

3.4. Delay-Line Initialization

Delay-line lengths strongly affect echo density and mixing time but are integer-valued and not continuously differentiable. Following Mezza et al. [13], we make them trainable as fractional delays through the frequency-sampling parameterization available in FLAMO. The delays are initialized on a logarithmic grid between 487 and 3361 samples (10 to 70 ms at $f_{\mathrm{s}} = 48$ kHz), and the fixed-delay variant freezes them at the co-prime integers $m =$ [487, 547, 631, 709, 809, 919, 1049, 1193, 1361, 1543, 1759, 2003, 2281, 2593, 2953, 3361]. Training the delays gives the lowest objective error, but they can drift toward near-commensurate values that color the late field. We discuss this limitation and its mitigation in Section 6.

3.5. Loss Function

Throughout this section, $y[n]$ denotes the target RIR and $\hat{y}[n]$ the model output. We write $X(\tau, k) = \mathrm{STFT}\{y\}(\tau, k)$ and $\hat{X}(\tau, k) = \mathrm{STFT}\{\hat{y}\}(\tau, k)$ for their short-time Fourier transforms at frame index τ and frequency bin k. We use a window length of 1024 samples with 50 % overlap.

3.5.1. Energy Decay Relief (EDR) loss

Per-octave-band EDC losses [7] give strong gradient signal for reverberation-time matching but quantize frequency into a small number of bands, limiting the feedback available to individual PEQ parameters. We instead operate on the Energy Decay Relief (EDR) [4], defined as the Schroeder backward-integrated power at every STFT bin

$$\mathrm{EDR}(t, k) = \sum_{\tau \geq t} |X(\tau, k)|^2. \tag{9}$$

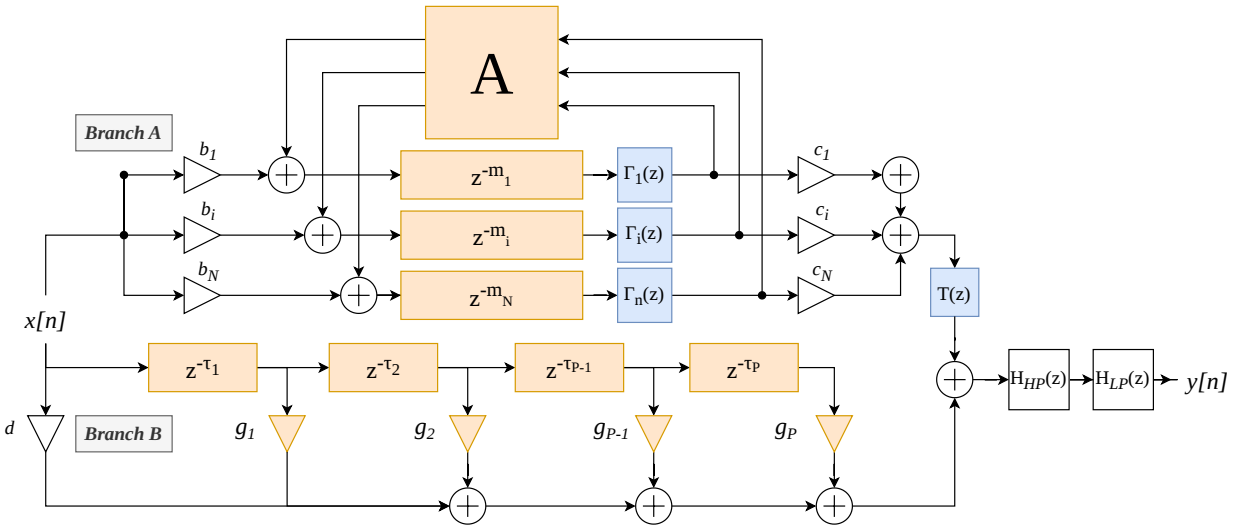


Figure 1: Block diagram of the proposed two-branch reverberator. Branch A (late reverberation) has input gains $\boldsymbol{b}$, the FDN recursion (feedback matrix, delays and scalable PEQ attenuation), output gains $\boldsymbol{c}$, and a post-loop tone filter $T(z)$. Branch B is the direct path and early-reflection tap-delay line. The summed branches pass through a trainable bandpass. Blue marks filters that can be PEQ or GEQ; orange marks parameters that can be trained or fixed (mixing matrix, delays, early-reflection taps). Section 5.4 ablates each component.

The model EDR $\widehat{\mathrm{EDR}}(t,k)$ is computed analogously from $\hat{X}$. Both surfaces are normalized to 0 dB at $t=0$ and converted to dB. Let $\varepsilon(t,k) = 10\log_{10}\mathrm{EDR}(t,k)$ and $\hat{\varepsilon}(t,k)$ the corresponding model surface. To restrict the loss to the dB range used by T_{30}, we apply a soft sigmoid mask

$$M(t,k) = \sigma\big(\beta\,(\varepsilon(t,k) - \varepsilon_{\mathrm{lo}})\big)\cdot\sigma\big(\beta\,(\varepsilon_{\mathrm{hi}} - \varepsilon(t,k))\big), \qquad (10)$$

with slope $\beta = 5$ and thresholds $\varepsilon_{\mathrm{hi}} = -5$ dB and $\varepsilon_{\mathrm{lo}} = -35$ dB. A soft mask, rather than a hard window, keeps the gradient continuous as EDR values cross the thresholds during training. The EDR loss is then the mean squared error in dB with $1/k$ frequency weighting

$$\mathcal{L}_{\mathrm{EDR}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{k}\sum_{t} M(t,k)\big(\hat{\varepsilon}(t,k) - \varepsilon(t,k)\big)^2. \qquad (11)$$

The $1/k$ weighting offsets the linear growth of octave bandwidth with k, so each octave contributes roughly equally and the PEQ parameters receive continuous frequency-domain supervision rather than the staircase signal of octave-band grouping.

3.5.2. *Band energy and power spectrum*

The band-energy loss $\mathcal{L}_{\mathrm{BE}}$ penalizes broadband spectral-tilt mismatch. It is the mean squared error, over octave bands f_c from 31.5 Hz to 16 kHz, between the log total energies $\log E_{f_c}(\hat{y})$ and $\log E_{f_c}(y)$ of the band-filtered signals. The power-spectrum loss $\mathcal{L}_{\mathrm{PS}}$ captures the spectral shape, especially useful for the early-reflections optimization. The time-summed power spectra $\sum_\tau |X(\tau,k)|^2$ of target and model are smoothed with a 16-bin running mean over k, converted to dB, and compared by a $1/k$-weighted squared dB difference.

3.5.3. *Auxiliary losses*

Auxiliary losses target room-acoustic features directly. The direct-to-reverberant ratio (DRR) loss is the squared dB-domain error between target and estimated DRR, with DRR defined as the energy ratio of samples within and beyond a 2.5 ms window after onset, which isolates the direct sound from the first reflections at typical source-listener distances, following the direct-sound convention in [22]. The early-energy (EE) loss is the mean squared dB error of the early-to-late energy ratios at 5, 50, and 80 ms, targeting clarity and the direct-to-reverberant balance. Four further terms sharpen decay and temporal structure. They are a per-octave-band T_{30} loss (squared error of the EDC slope fitted between -5 and -35 dB in each band), a linear-scale broadband EDC loss complementing the dB-scale EDR, an echo-density loss matching the echo-density profile [22], and a Gaussian-smoothed time-domain L_1 loss that gives the trainable delays and early-reflection taps a usable gradient.

The composite training loss is given by

$$\mathcal{L} = \sum_j \hat{\alpha}_j\,\mathcal{L}_j, \quad \hat{\alpha}_j = \min\Big(\frac{w_j}{\mathcal{L}_j^{(0)}},\ \alpha_{\max}\Big), \qquad (12)$$

where j ranges over the nine terms (EDR, band-T_{30}, linear-EDC, BE, PS, EE, DRR, echo-density, smoothed-time), w_j is the weight for term j, and $\mathcal{L}_j^{(0)}$ is its value at the initial model output for that RIR. Normalizing by $\mathcal{L}_j^{(0)}$ makes each term start at its target weight regardless of scale. The cap $\alpha_{\max} = 20$ stops a term that is already near-zero at initialization, such as the linear EDC when the PEQ is initialized from the target decay, from acquiring a runaway weight. The weights $(w_{\mathrm{EDR}}, w_{T_{30}}, w_{\mathrm{lin}}, w_{\mathrm{BE}}, w_{\mathrm{PS}}, w_{\mathrm{EE}}, w_{\mathrm{DRR}}, w_{\mathrm{ed}}, w_{\mathrm{st}}) = (2,5,3,2,2,3,2,2,1)$ were chosen by sweeping each term in isolation on a held-out RIR.

3.6. Training Details

We use Adam (learning rate 10^{-2}) for up to 300 epochs with early stopping. The FFT size is $1.2\times$ the target's broadband T_{30} (minimum 2 s), giving enough headroom to capture the reverberant tail without processing background noise. Targets are onset-aligned and peak-normalized, and each epoch is a single forward-backward pass on the (impulse, target) pair. Training takes one to a few minutes per RIR on a single NVIDIA RTX 4070 Super GPU and an Intel i7-14700K CPU.

The default configuration uses $N = 16$ delay lines and $B = 10$ PEQ bands. In Section 5.4 we evaluate the effect of varying the band count and compare with GEQ filters. Branch B uses $P = 64$ early-reflection taps (Section 3.2). To avoid overfitting to measurement noise, we inject noise matched to the target's noise floor (estimated from its final 100 ms) during training only, following Dal Santo et al. [23].

4. EVALUATION SETUP

4.1. Dataset

We evaluate on a curated set of nine measured RIRs from the OpenAIR dataset [24], spanning a wide range of acoustic conditions, from a small studio ($T_{30} \approx 0.2$ s), through theatres, lecture and concert halls, to a cathedral and a sports hall ($T_{30} \approx 6$ s). We selected the RIRs manually for a low noise floor across octave bands and the absence of truncation or audible artifacts. All RIRs are resampled to 48 kHz, trimmed after onset detection, and peak-normalized, and the FFT size for each target is auto-sized from its measured broadband T_{30}. The target RIRs are available in the repository linked in Section 1, and the released code reproduces every result and scales to larger corpora.

4.2. Baselines

We compare against a noise-shaped reverb baseline [25] implemented in PyTorch. It filters white noise through a 24-band octave filterbank (4093-tap FIR filters) and per-band exponential decay envelopes, giving 48 trainable parameters (24 gains and 24 decays) initialized from the target's per-band energy and decay. It is trained with the same composite loss as the FDN, but skips the first 200 ms of output to account for the filterbank's group delay. By construction it renders a statistically dense, stochastic late field from the onset rather than the discrete early reflections and direct sound of a measured room, a difference the structural metrics (DRR, early-energy ratios) make explicit. To make sure the baseline is not disadvantaged by our loss, we trained it with both its original multi-resolution STFT loss [25] and our composite loss, evaluated identically. The composite loss improves it on every metric, often by a large margin (the T_{30} MAE drops from 1.71 to 0.31 s, the MRSTFT distance from 684 to 80, and the EDR error from 13.3 to 7.8 dB), potentially because the multi-resolution STFT loss has no explicit decay term to guide the gradient, while our composite loss supplies a more convex search space. We therefore report the baseline throughout with our composite loss, the configuration that serves it best. It still relies on convolution, but is parametric and differentiable.

We also evaluate a differentiable FDN with trainable FIR attenuation and delay lines in the style of Mezza et al. [14]. With no public implementation available, we re-implemented the paper's version, drawing on parts of the public scattering-delay-network code [15] to stay close to the original. It is a 6-delay-line FDN running at 16 kHz. We attempted a 48 kHz extension but the results were inconclusive, so we keep the 16 kHz version. We refer to it as Mezza DFDN.

We also evaluate the RIR2FDN pipeline of Santo et al. [6], an analysis-synthesis approach whose analysis stage extracts the target's late-reverberation characteristics to set the FDN parameters and optimizes the feedback matrix as a scattering delay matrix of tiny delays and FIR filters. Using the authors' public code on our dataset, we keep their architecture, a 6-delay-line FDN with a two-stage attenuation filter, and refer to it as RIR2FDN. We tried to increase the number of delay lines to 16, but the optimization of the scattering delay matrix did not report results after several hours of training.

4.3. Metrics

Metrics are computed with the `pyfar` and `pyrato` packages rather than the differentiable proxies used in training, using Schroeder backward integration with noise-floor correction. Each reported value is the error between target and model, written as a loss term $\mathcal{L}$. The frequency-dependent metrics, T_{30} (reverberation time, -5 to -35 dB), EDT (early decay time, initial -10 dB slope), C_{80} (clarity, first-80 ms to remainder energy ratio), D_{50} (definition, first-50 ms energy fraction), and the early-energy ratios E_{50} and E_{80}, are evaluated independently in each octave band from 31.5 Hz to 16 kHz. Table 1 reports the mean absolute error across bands, averaged over the dataset, denoted $\mathcal{L}_{T_{30}}$, $\mathcal{L}_{\text{EDT}}$, $\mathcal{L}_{C_{80}}$, $\mathcal{L}_{D_{50}}$, $\mathcal{L}_{E_{50}}$, and $\mathcal{L}_{E_{80}}$. DRR, the echo-density profile (EDP), and the EDR error are broadband, denoted $\mathcal{L}_{\text{DRR}}$, $\mathcal{L}_{\text{EDP}}$, and $\mathcal{L}_{\text{EDR}}$. The EDP measures how reflection density builds up over time, computed from the kurtosis $\kappa(t)$ of a sliding window as $\text{EDP}(t) = \min(1,\, 3/\kappa(t))$, where 1 is fully diffuse and 0 fully sparse [22], and we report the mean absolute error of the EDP profile between target and model. The MRSTFT distance [25], denoted $\mathcal{L}_{\text{MRSTFT}}$, is a multi-resolution STFT distance at analysis scales of 256, 1024 and 4096 samples. For T_{30}, DRR, and EDR these share the name of the corresponding training loss in Section 3.5, as they measure the same quantity.

5. RESULTS

5.1. Objective Comparison

Table 1 reports the average training time, the total number of trainable parameter, the ISO 3382-1 [26] metrics, the EDR error (9), and a multi-resolution STFT distance for the three baselines and the proposed FDN family. Every frequency-dependent metric is the mean absolute error across octave bands.

Against the three baselines, the fully differentiable FDN, which jointly trains the delay lengths, mixing matrix, attenuation filters, and early-reflection taps, attains the lowest error on every metric except the EDR. It improves on the noise-shaped baseline in reverberation time (T_{30} MAE 0.261 vs. 0.311 s), clarity (C_{80} 2.87 vs. 3.42 dB) and definition (D_{50} 4.86 vs. 8.54 pp), and reduces the structural-fidelity errors by a wide margin. The direct-to-reverberant-ratio error is roughly ten times lower (0.52 vs. 5.28 dB) and the early-energy errors about three times lower (E_{50} 2.10 vs. 7.49). The FDN renders an explicit direct path and discrete early reflections, whereas a filtered-noise model is statistically dense from the onset. The DRR and early-energy columns quantify this difference.

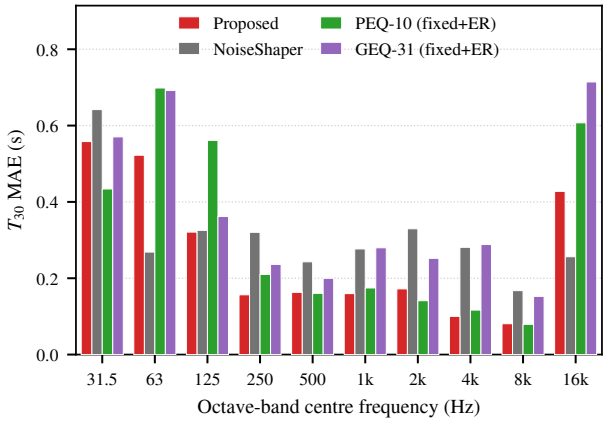


Figure 2: Per-octave-band T_{30} mean absolute error across the nine-RIR set, for the proposed FDN, the noise-shaped baseline, and the fixed+ER FDN with PEQ-10 and third-octave GEQ attenuation. The PEQ tracks every band as closely as the much larger third-octave GEQ; the error concentrates at the 31.5 Hz and 16 kHz extremes, where the target T_{30} estimate is noise-limited.

The noise-shaped baseline keeps the lowest EDR error overall (7.78 dB) and the lowest MRSTFT among the baselines (80.4) at a low training time. MRSTFT and EDR are perceptually relevant and outline that the noise-shaped baseline performs well to match the coloration of the late reverberation. Our informal listening confirmed it.

The Mezza-style differentiable FDN has the highest T_{30}, C_{80}, EDP, and EDR errors. Running at a 16 kHz sampling rate, it lacks the spectral resolution to match the target RIRs, but provides a reference point for the other fully differentiable approaches.

The RIR2FDN pipeline performs worst on the early-energy and direct-path metrics (D_{50}, E_{50}, E_{80}, DRR), yet is competitive on the EDR, where it edges out the proposed FDN (15.95 vs. 17.60 dB). Its analysis-synthesis loop captures the late decay well but does not render the early reflections and direct path. It is also the slowest method by a wide margin (about 10 minutes per RIR), as it optimizes a scattering feedback matrix of tiny delay lines and FIR filters and moves from PyTorch to MATLAB, which makes it unsuitable for an iterative differentiable pipeline.

Figure 2 breaks the T_{30} error down by octave band for the proposed FDN, the noise-shaped baseline, and the fixed+ER FDN with PEQ-10 and third-octave GEQ attenuation. The error is smallest in the mid-band and grows at the spectral extremes (31.5 Hz and 16 kHz), where measurement noise limits the target T_{30} estimate; the ten-band PEQ tracks every band as closely as the much larger third-octave GEQ.

5.2. Trained Bandpass Cutoffs

The trainable bandpass (Sec. 3.3), initialized at $f_{\text{lo}} = 100$ Hz and $f_{\text{hi}} = 18$ kHz, converges across the nine RIRs to a median f_{lo} of 11.2 Hz (range 3.4–44.1) and a median f_{hi} of 14.4 kHz (range 8.3–23.0). Because f_{lo} settles below 50 Hz on every target, the lowest octave bands (31.5–63 Hz) are left essentially un-attenuated, so the high-pass acts as a learnable rolloff guard rather than a fixed shelf, and the 4th-order high-pass does not bias their T_{30} estimates.

5.3. Computational Cost

Table 2 lists the per-component cost breakdown. The figures count multiplications per output sample only and do not capture the vector-pipelining or background-thread amortization that modern partitioned-convolution implementations exploit, so the comparison is a first-order indicator of arithmetic cost rather than an exact metric. At the example length shown (3 s RIR), partitioned convolution and the proposed PEQ FDN need comparable arithmetic per sample (∼1,222 vs. 1,155 multiplications), while the FDN uses roughly $11\times$ less persistent state memory, as it never stores the whole RIR. This memory advantage matters most for embedded and many-instance deployments (mobile devices, multichannel mixing consoles) rather than desktop hosts, and shows the architecture suits very constrained targets such as microcontrollers. Unlike partitioned convolution, the FDN is fully parametric. Once optimized, the learned PEQ prototype can reshape the virtual room's acoustics in real time. Training time scales with target T_{30}, since the FFT window is auto-sized to $1.2 \times T_{30}$. Short rooms ($T_{30} < 4$ s) converge in under two minutes, while longer rooms can take over twenty minutes for our largest (third-octave GEQ) model. PEQ models train substantially faster than GEQ models at the same band or parameter count.

5.4. Attenuation Filter Comparison

The lower half of Table 1 ablates the proposed FDN along two axes, all at $N = 16$ delay lines with a fixed Hadamard matrix, fixed delays, and early-reflection taps enabled. Varying the PEQ band count between 6, 10, and 16 changes T_{30} MAE by under 0.03 s. We select ten bands as the middle ground, judged on the EDR and the MRSTFT, PEQ-10 attains the lowest MRSTFT (45.4) and a low EDR (14.80 dB) of the three. Going to sixteen bands gives no meaningful improvement in frequency matching, since the EDR drops only marginally to 14.04 dB while the MRSTFT and T_{30} both worsen, and it nearly doubles the training time (5.2 vs. 2.8 min). At a matched FDN structure, PEQ-10 reaches a lower T_{30} error than both GEQ variants and uses $2.8\times$ fewer multiplications per output sample than the third-octave GEQ (966 vs. 2751 total multiplications, the difference being the attenuation filter) while training about $6\times$ faster (2.8 vs. 17.2 min).

5.5. Delay and Matrix Optimization

Making the entire model trainable lowers error. Training the mixing matrix and delay lengths on top of the fixed-Hadamard, fixed-delay model reduces T_{30} MAE from 0.315 to 0.261 s and D_{50} from 7.27 to 4.86 pp. The trained delays move only modestly, on average 14 samples (0.29 ms) and at most 53 samples (1.1 ms) from their co-prime initialization.

The EDP captures how reflection density grows from the onset. The noise-shaped reverb is statistically dense almost immediately, whereas our approaches, especially with optimized delay lines, achieve the lowest EDP error by matching the target's gradual buildup from sparse early reflections. The gain appears specific to delay training, since the trainable mixing matrix alone does not improve EDP. This confirms the observations of Mezza et al. [13]. The early-reflection taps also lower it, as they mimic the target's early reflections.

The early-reflection taps trade a little EDT (0.235 to 0.267 s with trained delays, as the taps fit early energy that competes with the late EDT slope) for a large gain on the early-energy and clarity

Table 1: Mean absolute error per metric, averaged over the 31.5 Hz to 16 kHz octave bands (DRR broadband) and across 9 RIRs. Train is the mean wall-clock time per RIR in minutes (RIR2FDN excludes its MATLAB render). Params is the number of trainable parameters; the fewest per block is in **bold**. Mults is real multiplications per output sample for the FDN recursion († NoiseShaper uses convolutional synthesis). For the metric columns, global best in **bold** and best within each block underlined. The proposed all-differentiable model is strongest against the three baselines; within the FDN family other rows can win individual columns (for example the no-ER variant on T_{30} and EDT). PEQ-10 (fixed+ER) and FDN (fixed) are the same configuration, repeated as the anchor of both ablation axes. ↓ lower is better.

Method	Train↓	Params	Mults↓	T_{30}↓	EDT↓	C_{80}↓	D_{50}↓	DRR↓	E_{50}↓	E_{80}↓	EDP↓	EDR↓	MRSTFT↓
Baselines													
NoiseShaper	**0.2**	**48**	†	0.311	0.269	3.42	8.54	5.28	7.49	7.19	0.135	**7.78**	80.4
Mezza DFDN	**0.2**	496	**490**	1.382	0.824	6.45	11.93	8.08	5.87	6.06	0.303	28.87	194.2
RIR2FDN	9.7	187	1175	0.444	0.877	5.96	16.55	15.15	16.70	16.98	0.216	15.95	273.7
Proposed FDN (PEQ-10)													
Proposed (all-diff)	1.6	494	1222	0.261	0.267	**2.87**	**4.86**	0.52	**2.10**	2.29	**0.076**	17.60	52.6
FDN (fixed delay, train mix)	1.4	478	1222	0.307	0.252	2.92	5.60	0.40	2.51	2.54	0.088	14.73	**44.6**
FDN (fixed)	2.8	222	966	0.315	0.378	3.67	7.27	0.44	2.29	2.41	0.092	14.80	45.4
FDN (fixed, no ER)	1.3	**95**	902	**0.245**	**0.230**	3.02	6.33	0.43	6.69	6.69	0.128	20.32	65.5
FDN (all-diff, no ER)	1.6	367	1158	0.246	0.235	3.09	5.77	0.45	5.93	5.87	0.077	20.12	60.8
Attenuation (fixed+ER)													
PEQ-6	1.1	**198**	626	0.313	0.385	3.53	6.85	0.52	2.44	2.68	0.102	15.77	45.5
PEQ-10	2.8	222	966	0.315	0.378	3.67	7.27	0.44	2.29	2.41	0.092	14.80	45.4
PEQ-16	5.2	258	1476	0.342	0.374	3.77	7.01	0.62	2.19	**2.08**	0.091	14.04	54.0
GEQ oct-10	2.0	204	966	0.334	0.376	3.85	7.21	0.41	2.62	2.43	0.089	15.66	72.5
GEQ 1/3-oct	17.2	240	2751	0.369	0.417	3.90	7.67	**0.24**	2.75	2.60	0.091	17.25	86.6

Table 2: Multiplications and state memory per output sample. FDN cost is T_{30}-independent, and convolution scales linearly with RIR length. Memory counts all persistent state (delay buffers, filter coefficients and state variables, matrix entries) needed for real-time sample-by-sample rendering. Partitioned convolution uses $L = 512$ ($\sim$11 ms latency at 48 kHz).

Component	Mult./sample	Memory (kB)
FDN core ($N = 16$)	256	95.52
Attenuation ($B = 10$)	800	4.38
Tone filter ($B = 10$)	50	0.27
Bandpass (HP+LP)	20	0.05
I/O gains	32	0.13
Early-reflection taps ($P = 64$)	64	0.50
Total proposed	**1,222**	**100.85**
Partitioned conv. (3 s) [2]	$\sim$1,155	$\sim$1,143

metrics. They cut E_{50} and E_{80} by about a factor of three (e.g. E_{50} from 5.93 to 2.10) and lower the EDR by a few and the MRSTFT by up to 20 points.

The direct-to-reverberant ratio is already low for every FDN variant, since the FDN renders an explicit direct path, so the taps leave it essentially unchanged. Fully training the delay lengths gives the best objective scores but can push the delays toward near-commensurate values that add a metallic, ringing coloration to the late field. Training only the mixing matrix recovers most of the gain without this artifact, but it colors the feedback matrix and trades away the orthogonality that keeps decay and coloration separate. Because the EDR stays very close between the fixed and trained-matrix variants and trainable delays raise the error only slightly, the fixed-delay, fixed-Hadamard variant is the safer deployment choice. More work is needed to understand how to train the delay lengths and matrix without introducing coloration.

6. CONCLUSION

We presented a gradient-descent method for matching measured room impulse responses with a fully differentiable feedback delay network. The proposed PEQ with $B = 10$ reaches a lower T_{30} error than octave and third-octave GEQ filters, while using $2.8\times$ fewer multiplications per output sample than the third-octave GEQ and training about $6\times$ faster.

An early-reflection tap-delay line is introduced to match the early energy of the target RIR, with trainable tap gains and positions, while the late-reverberation delay lengths and feedback matrix are also trainable. Making these components trainable greatly reduces the error. The early-reflection taps cut the early-energy error E_{50} from 5.93 to 2.10, and training the delay lengths and mixing matrix lowers the T_{30} error from 0.315 to 0.261 s and D_{50} from 7.27 to 4.86 pp. Our composite loss, although it combines several terms, covers the temporal and spectral aspects of the RIR that should be matched.

Against three baselines, a noise-shaped reverberator [25], a Mezza-style differentiable FDN [13], and the analysis-synthesis RIR2FDN pipeline [6], the fully differentiable FDN attains the lowest error on every metric except the EDR. Training the noise-shaped baseline with our composite loss in place of its original multi-resolution STFT loss improves it on every metric, which indicates that the loss, and not only the architecture, drives the gains. The noise-shaped baseline scores best on the EDR, and our informal listening suggests it is perceptually close to the target, so the EDR and the multi-resolution STFT distance may be perceptually relevant, which we plan to confirm in a formal listening test.

The main limitation is coloration when training the delay

lengths. It gives the lowest objective error but can add a metallic ringing to the late field. The fixed-delay variant avoids this issue at a small cost in objective error, so it is the safer choice for deployment. More investigation is needed to understand how to train delay lines without introducing coloration. The current system is mono. Extending it to stereo or multichannel output through per-channel gains or a short binaural FIR is a natural continuation, as is neural parameter prediction that removes the per-RIR gradient descent altogether.

7. REFERENCES

[1] A. Farina, "Advancements in Impulse Response Measurements by Sine Sweeps," in *122nd Audio Engineering Society Convention*, Vienna, Austria, 2007.

[2] W. G. Gardner, "Efficient Convolution without Input/Output Delay," *Journal of The Audio Engineering Society*, vol. 43, pp. 127–136, 1995.

[3] J.-M. Jot and A. Chaigne, "Digital Delay Networks for Designing Artificial Reverberators," in *90th Audio Engineering Society Convention*, Paris, France, Feb. 1991.

[4] J. M. Jot, "An analysis/synthesis approach to real-time artificial reverberation," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2, San Francisco, CA, USA, 1992, pp. 221–224.

[5] I. Ibnyahya and J. D. Reiss, "A Method for Matching Room Impulse Responses with Feedback Delay Networks," in *153rd Audio Engineering Society Convention*, New York, USA, Oct. 2022.

[6] G. D. Santo, B. Alary, K. Prawda, S. Schlecht, and V. Välimäki, "RIR2FDN: An improved room impulse response analysis and synthesis," in *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24)*, Guildford, UK, Sep. 2024, pp. 230–237.

[7] G. D. Santo, K. Prawda, S. Schlecht, and V. Välimäki, "Differentiable Feedback Delay Network For Colorless Reverberation," in *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx23)*, Copenhagen, Denmark, Sep. 2023, pp. 244–251.

[8] V. Välimäki, K. Prawda, and S. Schlecht, "Two-Stage Attenuation Filter for Artificial Reverberation," *Signal Processing Letters, IEEE*, vol. 31, pp. 391–395, Jan. 2024.

[9] G. D. Santo, G. M. De Bortoli, K. Prawda, S. J. Schlecht, and V. Välimäki, "FLAMO: An Open-Source Library for Frequency-Domain Differentiable Audio Processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hyderabad, India, Apr. 2025, pp. 1–5.

[10] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," in *8th International Conference on Learning Representations*, Online, 2020.

[11] S. Lee, H.-S. Choi, and K. Lee, "Differentiable Artificial Reverberation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2541–2556, 2022.

[12] L. Rabiner, B. Gold, and C. McGonegal, "An approach to the approximation problem for nonrecursive digital filters," *IEEE Transactions on Audio and Electroacoustics*, vol. 18, no. 2, pp. 83–106, Jun. 1970.

[13] A. I. Mezza, R. Giampiccolo, E. De Sena, and A. Bernardini, "Data-driven room acoustic modeling via differentiable feedback delay networks with learnable delay lines," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2024, no. 1, p. 51, Oct. 2024.

[14] A. I. Mezza, R. Giampiccolo, and A. Bernardini, "Modeling the Frequency-Dependent Sound Energy Decay of Acoustic Environments with Differentiable Feedback Delay Networks," in *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24)*, Guildford, UK, 2024, pp. 238–245.

[15] A. I. Mezza, R. Giampiccolo, E. De Sena, and A. Bernardini, "Differentiable Scattering Delay Networks for Artificial Reverberation," in *Proceedings of the 28th International Conference on Digital Audio Effects (DAFx25)*, Ancona, Italy, Sep. 2025.

[16] J. M. Jot, "Proportional parametric equalizers - Application to digital reverberation and environmental audio processing," in *139th Audio Engineering Society International Convention*. New York, USA: AES, Oct. 2015.

[17] I. Ibnyahya and J. D. Reiss, "Differentiable Attenuation Filters for Feedback Delay Networks," in *Proceedings of the 28th International Conference on Digital Audio Effects (DAFx25)*, Ancona, Italy, Sep. 2025.

[18] S. J. Schlecht and E. A. P. Habets, "Accurate reverberation time control in feedback delay networks," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Aug. 2017, pp. 337–344.

[19] S. Nercessian, "Neural Parametric Equalizer Matching Using Differentiable Biquads," *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx)*, pp. 265–272, Sep. 2020.

[20] J. A. Moorer, "About This Reverberation Business," *Computer Music Journal*, vol. 3, no. 2, pp. 13–28, 1979.

[21] R. Bristow-Johnson, "Audio-EQ-Cookbook," 2021, available at "https://www.w3.org/TR/audio-eq-cookbook/" accessed June 08, 2025.

[22] J. S. Abel and D. P. Berners, "A technique for nonlinear system measurement," in *121st Audio Engineering Society Convention*, vol. 3, California, USA, 2006, pp. 1057–1063, iSBN: 9781604236637.

[23] G. D. Santo, K. Prawda, S. J. Schlecht, and V. Välimäki, "Learning Recursive Attenuation Filters Under Noisy Conditions," Dec. 2025, arXiv:2512.16318 [eess].

[24] D. T. Murphy and S. Shelley, "OpenAIR: An Interactive Auralization Web Resource and Database," *Journal of the Audio Engineering Society*, no. 8226, Nov. 2010.

[25] C. J. Steinmetz, V. K. Ithapu, and P. Calamia, "Filtered Noise Shaping for Time Domain Room Impulse Response Estimation from Reverberant Speech," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Online, Oct. 2021, pp. 221–225.

[26] "Acoustics - Measurement of room acoustic parameters — Part 1: Performance spaces, ISO 3382-1," 2009, issue: 3382-1.