Daniotti *et al.*
EURASIP Journal on Audio, Speech, and Music Processing (2026) 2026:14
https://doi.org/10.1186/s13636-025-00442-1

Journal on Audio,
Speech, and Music Processing

EMPIRICAL RESEARCH

Open Access

Tilt loss: a perceptual loss function to improve music packet loss concealment

Filippo Daniotti[1*], Luca Vignati[1] and Luca Turchet[1]

Abstract

Recent advancements in deep learning have paved the way for novel approaches to the problem of Packet Loss Concealment (PLC) in networked music performance systems. However, deep neural networks may have large inference times and, therefore, violate the strict temporal requirements of PLC methods for such systems. A promising avenue in this space lies in the exploration of the loss function used to train the network. Indeed, loss functions have a direct impact on the latent representation learned by the model during the training process without any additional cost at inference time. In this paper, we present the Tilt Loss, a perceptual loss function, i.e., a loss function that allows the model trained with it to have performances that correlate with the human evaluation. The proposed method was able to outperform the current state-of-the-art in PLC methods according to human evaluation, albeit the model exhibited unsatisfactory performance with unpitched instruments. Furthermore, our study pinpoints the need for novel objective metrics specifically tailored for the PLC case.

Keywords Internet of musical things, Networked music performance, Packet loss concealment, Deep learning, Neural network

1 Introduction

Real-time communication over packet-switched networks has acquired an ever-increasing importance in today's society, prompted by events such as the recent SARS-CoV-2 pandemic, and this trend is not reverting in the near future. The widespread adoption of the internet has extended its reach to physical objects, a concept known as the Internet of Things. This transformation also affected the music domain, giving rise to the Internet of Musical Things (IoMusT) [1], which connects musical instruments, audio devices, and other music-related objects to the internet.

A central component of the IoMusT is represented by Networked Music Performance (NMP) systems, which aim at enabling geographically displaced musicians to perform together synchronously over the internet [1]. However, NMP poses stringent latency requirements, as the quality of the musical interaction would rapidly degrade when the latency exceeds 30 ms, as shown by different studies in the review reported in [2]. To fulfill such latency constraints, best-effort protocols based on User Datagram Protocol (UDP) are typically used in such applications ([2], V.D). By prioritizing latency over reliability, such protocols may cause packets to be lost during the transmission, introducing playback artifacts and, consequently, a noticeable degradation in the audio quality at the receiver's side. This creates a need for algorithms that conceal information in lost packets; such a class of algorithms is referred to as Packet Loss Concealment (PLC) algorithms ([2], IV.D) [3].

While many traditional approaches to PLC exist [3], including linear autoregressive models [4, 5], novel approaches leverage the recent developments in deep learning techniques [6, 7]. However, deep learning models usually have high inference times due to their size, limiting real-time usage. Hardware acceleration offers a solution, but it is not always feasible for consumer

*Correspondence:
Filippo Daniotti
filippo.daniotti@unitn.it
[1] Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 9, 38123 Trento, Italy

devices used in the IoMusT. A promising avenue for models capable of higher-quality concealments lies in the exploration of the loss function used to train the model. Loss functions provide a quantitative basis for the underlying optimization algorithm to operate and update the model weights during the training process. The crucial advantage of using loss functions as subjects for this study is the fact that they do not add any contribution to the inference time, irrespective of their computational complexity, as they solely influence the offline training phase.

Based on all the above, in this study we propose and assess the Tilt Loss, a perceptual loss function, which, similarly to [8], is a loss function that allows a deep learning-based PLC model trained with it to have performances that correlate with human perceptions. We base our PLC approach on PARCnet [7], specifically reusing its neural network architecture as our backbone. To adapt it to our framework, we retrain the model using the Tilt Loss function instead of the original loss formulation. To the best of our knowledge, this is the first attempt to utilize a perceptually motivated loss function to the problem of PLC in the context of NMP. This work has been conducted in the context of IEEE IS^2 2024 PLC Challenge and later submitted to the same challenge [9].

2 Background

2.1 Packet loss concealment

PLC methods in NMP applications are algorithms employed to mitigate the effects of packet loss in audio streams [3, 4]. Packet loss occurs when data packets are lost or delayed too much during transmission over a network. This may be due to different factors such as network congestion, network errors, or high jitter [2]. In the context of NMP applications, packets are usually not retransmitted, as the retransmission delay would introduce unacceptable latencies and increase the bandwidth ([2], V.D). Therefore, PLC algorithms are employed in order to *conceal* the lost frames in the incoming audio stream.

PLC algorithms can be classified into two main categories: model-based and waveform substitution [4]. Model-based PLC algorithms use a model to predict the missing packets based on the valid context. Typically, model-based PLC have employed linear models, such as an autoregressive (AR) model [5, 10] ([4], 2.1); recently, advancements in deep learning caused non-linear model to be employed, specifically neural networks.

From the machine learning engineering standpoint, PLC can be described as a supervised learning task with structured output; typically, it is approached as a time-series forecasting regression task. Audio recordings are time series usually encoded in PCM as either 16- or 24-bit integers; however, in audio processing tasks, signals are usually normalized to the range $[-1, 1]$ as 32 or 64-bit precision floating point numbers. Hence, the overall objective is to learn the parameters θ of an approximator function

$$f_\theta : [-1, 1]^{\mathcal{N}} \rightarrow [-1, 1]^{\mathcal{M}}$$

that maps a *context* $\mathbf{x}$ to the concealed packet $\hat{\mathbf{y}}$ in order to minimize the loss function between the ground truth $\mathbf{y}$ and the prediction $\hat{\mathbf{y}}$.

$$\theta = \arg\min_{\theta} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}), \quad \hat{\mathbf{y}} = f_\theta(\mathbf{x})$$

The context $\mathbf{x}$ encompasses a buffer of ν valid packets of N samples each, yielding a total length of $\mathcal{N} = \nu \cdot N$ samples. However, the specific information carried by the context vector is model-dependent, as some approaches uses a context vector of raw audio samples [7] and others employ a time-frequency domain representation of the signal [6]. The prediction of the model $f_\theta(\mathbf{x}) = \hat{\mathbf{y}}$ represents the N samples of the predicted packets. Some approaches [7] ([4], 2.1) also predict M additional samples to be cross-faded with the next incoming valid packet, in order to prevent discontinuities, yielding a total prediction length of $\mathcal{M} = N + M$ samples.

2.2 Related work

A first attempt to use deep learning to address PLC specifically in the context of NMP was presented in Verma et al. [6]. The proposed solution consists of a double-branched neural network model to learn a representation from the valid context, combining both the time and the time-frequency domain. It uses a loss function that measures the absolute difference between the predicted and ground truth audio packets, with a reweighting mask applied sample-wise to penalize prediction errors for samples located at the edges of a packet more than for those in the center. This is implemented by multiplying the absolute error of each sample by an inverse Hanning window.

Mezza et al. [7] followed a different approach. They proposed PARCnet, a hybrid method combining autoregressive modeling and deep learning. Such a method uses a 1D Convolutional Neural Network with dilation and residual connections to model the residual of a linear AR predictor. Hence, the model has two independent branches, one being a AR model, the model of which is fitted at runtime, and the other being the neural network model. The role of the neural network model is to predict the *residual* of the AR prediction. Equation (1) reports the mathematical definition of PARCnet.

$$\begin{cases} \hat{y}[n] = y_{\mathrm{AR}}[n] + f_\theta(\mathbf{x})_{n-k}, \\ y_{\mathrm{AR}}[n] = \sum_{i=1}^{\rho} \varphi_i y_{\mathrm{AR}}[n-i], \end{cases} \tag{1}$$

Considering an audio packet of length M samples and $n = k, \ldots, k+M-1$, k is the index of the first missing sample, $\hat{y}[n]$ is the predicted sample, $y_{\mathrm{AR}}[n]$ is the prediction of the AR component, $f_\theta(\mathbf{x})$ is the neural network component, ρ is the number of past samples used by the AR component and φ_i encompass the parameters of the AR component.

The method considers packets of length $N = 320$ samples sampled at 32kHz. Additionally, the model predicts $M = 80$ additional samples to be linearly cross-faded with the following packet, yielding a total prediction size of $\mathcal{M} = 400$ samples. The model was benchmarked to have an average inference time of around 8 ms. PARCnet uses a backbone AR model of order $\rho = 128$. It considers a context buffer of 10 packets and its parameters are computed with the autocorrelation method [11] using the Levinson-Durbin algorithm [12]. The deep learning model is a lightweight 1D CNN of 416 k parameters. It features three main components: an encoder, a bottleneck, and a decoder. These components work together to process and transform the input data through the network.

PARCnet's loss function features a time domain component and a time-frequency domain component. While the time domain component is the simple Mean Squared Error, this loss has two time-frequency components: the spectral convergence loss $\mathcal{L}_{sc}^{(q)}$ and the log spectral loss $\mathcal{L}_{log}^{(q)}$. These two losses were already used in waveform synthesis ([13], III) and in PLC applied to speech ([14], III.D). The loss is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{MSE}} + \frac{\lambda}{Q} \sum_{q=1}^{Q} \left(\mathcal{L}_{sc}^{(q)} + \mathcal{L}_{log}^{(q)} \right) \tag{2}$$

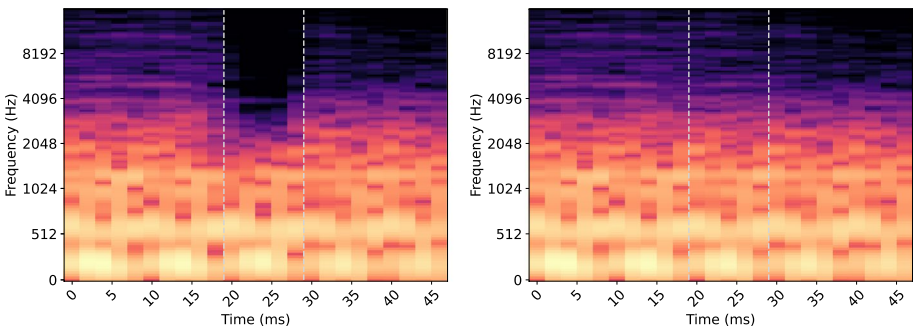
where Q encompasses different sets of parameters of the FFT computation and $\mathcal{L}_{\mathrm{MSE}}$ is the mean squared error (MSE) loss on the time domain.

PARCnet was evaluated over a series of qualitative and quantitative metrics against several baselines, including other deep-learning methods and AR models. It outperformed by a wide margin every other method on every metric, and as of today, it is the state-of-the-art model for this task. Yet, the algorithm has an average inference time of over 8 ms and it works with a 10-ms packet size. By contrast, state-of-the-art AR models employed in commercial settings utilize smaller packet sizes of 2.6 ms (128 samples at 48 kHz [15]), achieving inference times as low as 1 ms.

3 The tilt loss function

The motivations underlying the design of a perception-based loss function are rooted in empirical listening tests on the prediction data from the PARCnet model performed by the authors. These tests highlighted the fact that the concealed signal was unbalanced in the reconstruction of the frequency content, as frequencies in the low and the mid-frequency range were more accurately reconstructed, while frequencies in the high range were noticeably more inaccurate. This behavior can be clearly seen from the spectrograms of the concealments (Fig. 1).

Hence, the idea of rebalancing the frequency spectrum in favor of the high-frequency range arose in order to push the model to focus thoroughly on learning a better representation of the high-end. This rebalancing is defined as a reweighting mask applied to the



(a) Spectrogram, Baseline prediction. (b) Spectrogram, ground truth.

Fig. 1 Mel spectrograms representing the packet starting at sample index 2128000 from the validation set of the Bach Cello Suite dataset. The packet spans from millisecond 19 to 29 in the displayed spectrograms. The spectrograms were computed with STFT parameters: FFT size 2048, hop length 64, and window length 256. The concealment from model Baseline (left) is significantly less precise in the reconstruction of the high-end compared to the ground truth (right). This can be seen from both the spectrogram of the Baseline model prediction, which features a hole in the image where the high-end of the concealment should be

frequency bins of the mean absolute error (MAE) of the predicted and the ground truth, hence acting as an upward tilt filter.

The loss is defined as follows:

$$\mathcal{L}_{\text{TILT}}^{(i)} = \frac{1}{T \cdot F} \sum_{t \in T} \sum_{f \in F} \left| \vartheta\left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right)_f \cdot \left(|\hat{\mathcal{S}}|_{t,f}^{(mel)} - |\mathcal{S}|_{t,f}^{(mel)} \right) \right| \tag{3}$$

T is the number of time frames; F is the number of mel bins; i is the index of the packet; $\hat{S}[t,f]^{(mel)}$ is the mel spectrogram of the predicted signal; $S[t,f]^{(mel)}$ is the mel spectrogram of the ground truth signal; ϑ is the *Tilt frequency reweighting function* and $\vartheta\left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right)_f$ is the coefficient of the *spectral adaptive reweighting mask* for frequency bin f.

The spectral adaptive reweighting mask is defined through the Tilt frequency reweighting function ϑ. The spectral adaptive reweighting mask—shorthanded as *reweighting mask* from now on—is *dynamic* with respect to the frequency content of the context window $\mathbf{x}^{(i)}$. This is crucial for musical audio, as the frequency content of the signal varies a lot depending on the instrument, style, or even individual passage.

Algorithm 1 summarizes the computation procedure of the reweighting mask in pseudocode, with some elements taken from the PyTorch APIs.

Algorithm 1 Tilt frequency reweighting function ϑ computation

Data: Context $\mathbf{x}^{(i)}$, Ground truth $\mathbf{y}^{(i)}$, shape of the spectrogram
Result: Spectral adaptive reweighting mask

spectrum $\leftarrow$ `dft(concat(`$\mathbf{x}^{(i)}, \mathbf{y}^{(i)}$`)).abs()` ;

mel_spectrum $\leftarrow$ `MelFilterbak(`spectrum`)` ;

db_spectrum $\leftarrow$ `AmplitudeToDB(`mel_spectrum`)` ;

mask $\leftarrow$ `LowPassFilter(`db_spectrum`)` ;

mask $\leftarrow$ `Normalize(`mask, $[0, 1]$`)` ;

mask $\leftarrow 1 -$ mask ;

mask $\leftarrow$ mask.`Expand(`shape`)` ;

return mask

In the following, we present a detailed description of each operation in the computation of the reweighting mask.

First, the Discrete Fourier Transform (DFT) of the concatenation of the context and the ground truth is computed. The DFT is obtained through a FFT with a window size $\mathcal{M}$, which is the context length plus the packet length, and an FFT size equal to the next power of 2 of the window size, that is $2^{\lceil \log_2(\mathcal{M}) \rceil}$.

Next, the magnitude of the DFT result is transformed into the mel scale. The mel scale transformation uses a 128-bin mel filterbank.

Then, the mel spectrum is transformed to the dB scale as $20 \cdot \log_{10}\left(\mathcal{S}^{(\text{mel})} + \epsilon\right) + c$, where $\mathcal{S}^{(\text{mel})}$ denotes the mel spectrum, $\epsilon = 10^{-7}$ is a constant introduced to prevent numerical errors in the logarithm computation, and $c = 80$ is a constant used to shift the values of the curve.

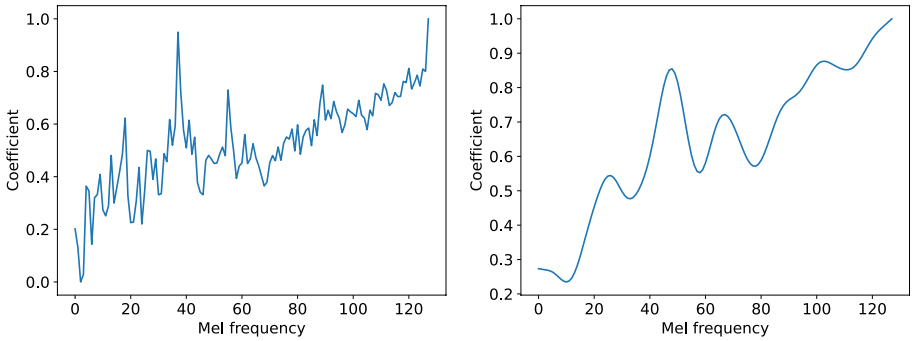
A low-pass filter is then applied to the mel spectrum to reduce high-frequencies from the spectrum, ensuring a gradual transition of the reweighting coefficients across frequency bins and encouraging more stable behaviour from the model. The filter is designed as a Butterworth [16] filter of order 5, with a cut-off frequency of 50 Hz. In order to prevent edge artifacts, we left pad the input by repeating the first value. After the filter is applied, the left pad is sliced from the output. The size of the left padding is equal to 50. The impact of the low-pass filtering on the resulting curve can be seen in Fig. 2.

The reweighting mask is normalized to the range $[0, 1]$. After that, the complement of the mask so far is taken. Finally, the mask is expanded to match the dimensions of the difference between the spectrogram of the prediction and the ground truth. Specifically, the mask is a 1D vector of size $[F]$, where F is the number of mel bins, and is expanded by repetition to $[B, T, F]$, where B is the batch size and T is the number of time bins.

4 Results

4.1 Data

Three datasets were used to assess our work: the *Bach Cello Suites* dataset [17], the *MAESTRO* dataset [18], and the *E-GMD* dataset [19]. Specifically, In this study, we use only the recordings from the "2006" subset of the MAESTRO dataset—from now on, referenced as Maestro (2006)—and the "drummer10" subset of the E-GMD dataset - E-GMD (drummer10). This choice was



(a) Reweighting mask, omitting the filtering. (b) Reweighting mask, with filtering.

Fig. 2 This figure demonstrates the effect of low-pass filtering in the computation of the reweighting mask. Here, we take one datapoint from the Bach Cello Suite dataset. The left plot depicts the mask computed without low-pass filtering, representing an ablated version of the full method. In contrast, the right plot shows the mask generated using the complete computation pipeline. The mel frequency bins are shown along the *x*-axis, with reweighting coefficients on the *y*-axis

motivated by computational resource constraints, as well as a desire to maintain a dataset size comparable to that of the Bach Cello Suites.

While most works on PLC feature models trained and evaluated on a single dataset [6, 7, 20–23], we chose to conduct the experiments on multiple datasets to ensure the robustness of the results. In fact, different musical instruments may produce radically different waveforms. These can be described by a number of parameters, such as envelope (tied to the evolution of sound dynamics over time), dynamic range, timbre (tied to the frequency distribution of the signal's power spectrum), and type of excitation (monophonic vs polyphonic, pitched vs unpitched). For this reason, the domain of musical audio involves a higher degree of variety than the domain covered in similar tasks (for instance, PLC applied to speech). Hence, using multiple datasets allows us to better assert the capability of the models - Baseline and Tilt Loss - to represent the musical domain.

In order to create the individual datapoints to use in the training procedure, the audio recordings are split into fixed-length packets. The packet length, in compliance with the method adopted in our backbone architecture [7], is set to 320 samples, which corresponds to 10 ms of audio at the target sampling rate—that is, 32 kHz. Audio recordings usually feature millions of samples, which would result in very large disk and memory requirements. Hence, to reduce the overall dimension of the problem, we produced a subset of the original dataset by skipping some of the candidate packets by a fixed value. We called such a value the packet skip factor (PSF); it acts as a lag variable that indicates the distance in packets between the index of the first sample of a packet and the index of the first sample of the following. The PSF used differs depending on the dataset and it is meant for the datapoint number to match across all of them. The value chosen and its implications are shown in Table 1.

Each datapoint features the following content: $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)}_{\text{AR}}\}$, where $\mathbf{x}^{(i)}$ is the context for the prediction and is a vector of size $[1, 2640]$, where 2640 is given by a context window of 7 packets plus a zeros of size 320 for the lost packet, $\mathbf{y}^{(i)}$ is the ground truth, i.e. the actual lost packet, and $\hat{\mathbf{y}}^{(i)}_{\text{AR}}$ is the prediction of the lost packet computed by the autoregressive component of PARCnet. The prediction of the AR model was computed offline within the data preparation pipeline.

The datasets were split at the recordings level, so on a per-file basis. A variation of the *hold-out* splitting strategy was adopted, where one file is held as a validation set, to be used to control overfitting and hyperparameter fine-tuning, one file is held out as a test set for evaluation, and all the remaining files are used as a training set.

Training setup All the training experiments were run on a 64-bit Linux machine running Ubuntu 22.04.3 LTS. The machine was equipped with an Intel(R) Core(TM) i9-10940X 3.30 GHz CPU with 2 threads per core and 14 cores, 200 GB of RAM, and two GPUs, namely an

Table 1 PSF—the distance in packets between the first samples of two consecutive datapoints in an audio track—used for the three datasets and the total number of datapoints

Dataset	PSF	Percentage	No. datapoints
Bach Cello Suite	10	$\propto 10\%$	337,834
MAESTRO (2006)	30	$\propto 3.3\%$	303,552
E-GMD (drummer10)	10	$\propto 10\%$	304,418

Percentage stands for the percentage of the total number of non-overlapping candidate packets within all the tracks in the datasets, with reference to a PSF value of 1

NVIDIA GeForce RTX 4090 and an NVIDIA GeForce RTX 3090, both featuring 24 GB of dedicated VRAM. All the experiments were performed with Python 3.11.5 and CUDA 12.2, leveraging the deep learning framework PyTorch [24] and its convenient wrapper Pytorch Lightning [25]. All the random generators were feeded with the same seed (42).

The RAdam optimisation algorithm [26] with $\beta_1 = 0.5$ and $\beta_2 = 0.9$ is employed as in the PARCnet paper [7]. Each model was trained for 100 epochs with a batch size of 64. The learning rate η was set to 10^{-3} and was decayed by a factor of 0.1 when the validation loss did not improve for a number of epochs; this patience value was set to 10.

4.2 Objective metrics

Traditionally, Packet Loss Concealment methods were evaluated using regular metrics for regression tasks. Those include the mean absolute error (MAE) [6], mean squared error (MSE) [10], and normalized mean squared error (NMSE) [7]. However, providing a measure of how well the model performs in terms of audio quality is not as straightforward as computing a mathematical difference in the reconstructed signals. In fact, the primary purpose of a PLC algorithm is not to reconstruct the original signal, rather to provide a concealed signal that is perceptually similar to the original signal, with minimal playback artifacts and disruption of the quality of the interaction. A number of objective metrics have been proposed to include perceptual elements in the domain of audio. However, all of these metrics are inadequate for the specific task of PLC applied to NMP, either because they are meant to encompass specific features relevant in other domains—e.g., PESQ [27], STOI [28] - or because they were tailored for different tasks—e.g., PEAQ [29], still widely used for PLC in NMP but proven to be ineffective [30]—or because they are model-based metrics designed for very specific contexts—e.g., PLCMOS [31].

Currently, there is a lack of a good metric tailored to the specificities of PLC in NMP. Hence, the only reliable way to assert the performance of a PLC algorithm is to use human evaluation. Several specifications for listening tests exist, such as the ITU. P.800.1 Mean Opinion Score (MOS) [32] or the ITU. BS.1534 MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) [33] test. While listening tests ensure robust results, they are time-consuming and intrinsically error-prone.

In this work, we corroborate objective metrics with subjective metrics. In terms of objective metrics, the MAE and the NMSE are considered in this work. Recall the definition of the MAE:

Table 2 Baseline vs. proposed loss function objective metrics results

Dataset	Loss	MAE	NMSE
Bach Cello Suites	**Baseline**	**1.177**	**−8.832**
	Tilt Loss	1.397	−7.353
MAESTRO	**Baseline**	**8.636**	**−4.268**
	Tilt Loss	9.295	−3.610
E-GMD	**Baseline**	**2.215**	**−5.953**
	Tilt Loss	3.670	−0.661

Baseline is PARCnet trained using the loss from [7]. Lower is better

Bold indicates the best performing model for each Dataset

$$\mathrm{MAE}^{(i)} = \frac{1}{N}\sum_{n=1}^{N}\left|\hat{y}[n]^{(i)} - y[n]^{(i)}\right|_1 \tag{4}$$

Likewise, the NMSE is defined as in [7]:

$$\mathrm{NMSE}^{(i)} = 10\log_{10}\frac{\|\hat{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)}\|_2^2}{\|\mathbf{y}^{(i)}\|_2^2}, \tag{5}$$

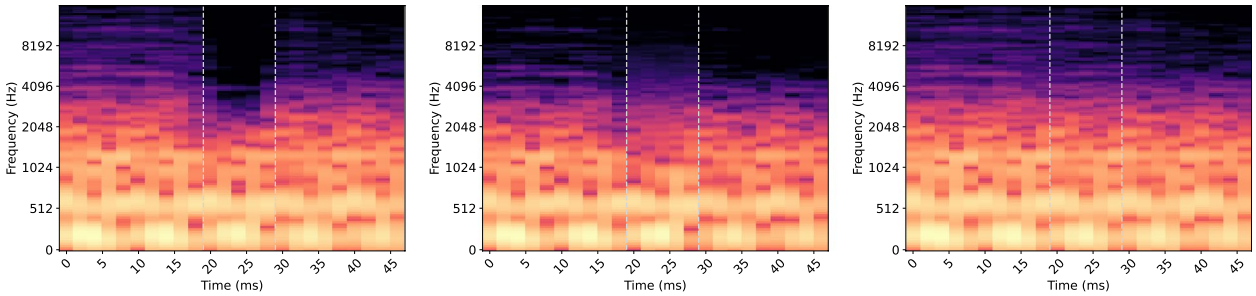
The results of the evaluation with objective metrics are presented in Table 2. Specifically, we compare Baseline (that is, PARCnet model trained using the Multi-Resolution STFT Loss function from [7]) against Tilt Loss, that is the same PARCnet model trained using the Tilt Loss function. We do not compare our solution against other state-of-the-art PLC algorithms, as the comparison in [7] still holds. From Table 2 it is possible to notice that the baseline method outperforms the proposed method according to both MAE and NMSE, in all of the three datasets.

Additionally, we provide an example of inference of both Baseline and Tilt Loss in Fig. 3.

4.3 User study

The goal of the listening test was to evaluate the quality of the audio generated by the proposed method with greater reliability than what objective metrics can guarantee. Participants were asked to rate the audio quality of audio clips that included concealed packets. The study was designed in compliance with the ITU-R BS1534-3 MUSHRA [33] specifications. The apparatus comprised a laptop delivering the sound stimuli through an external soundcard; the participants received the stimuli through professional-grade studio headphones (Beyerdynamic DT 770 Pro). The study was handled via the webMUSHRA listening test tool [34]. Specifically, the pyMUSHRA[1] wrapper application was used to facilitate results collection and analysis.

[1] https://github.com/nils-werner/pymushra



(a) Baseline prediction. (b) Tilt Loss prediction. (c) Ground truth.

Fig. 3 Mel spectrograms of the packet with start at sample index 2128000 from the validation set file (`ttue1.wav`) of the Bach Cello Suite dataset. The packet spans from millisecond 19 to 29 in the displayed spectrograms. The spectrograms were computed with STFT parameters: FFT size 2048, hop length 64, and window length 256. Notice that the prediction yielded by the model trained with the Tilt Loss has a better reconstruction of the high-end of the frequency spectrum

Participants A total of 15 musicians (13 males, 2 females, mean age 32.26 years, standard deviation 9.77 years) took part in the experiment. They had different levels of musical expertise, ranging from intermediate to advanced (average musical expertise = 23 years, standard deviation = 10 years). Each musician took on average around 28 min to complete the experiment.

Stimuli Nine audio clips were taken from the held-out test sets, three excerpts for each of the three datasets employed. The audio clips were selected through informal listening and aimed to represent a variety of playing styles, dynamics, and articulations. Similarly to [7], each clip was 10 s long. For each clip, a trace was generated by selecting a set of evenly spaced packets to be lost, where the space between the index of the first sample of each missing packet and the next is 100 ms. The trace was then used to generate the lost packets.

For each clip, 6 types of stimuli were created:

- *Reference*: the original audio clip without any modification;
- *Zero-filling*: missing samples in lost packets are zeroed out, acting as an *anchor*;
- *Autoregressive*: the lost packets are concealed using an AR model; specifically, we isolate the AR component of PARCnet [7] and slightly improve it by scaling up the order ρ to 512;
- *Baseline*: the lost packets are concealed using the baseline method, i.e., the PARCnet model ([7], II.B), retrained by the authors; the order of the AR component is $\rho = 128$;
- *Tilt Loss*: the lost packets are concealed using PARCnet trained with the Tilt Loss (Eq. (3)); the order of the AR component is $\rho = 128$.

We scale up the order ρ of the AR model to test both Baseline and Tilt Loss against a stronger linear model. It is not our intention to perform an ablation study on PARCnet, as such study is present in the PARCnet's paper [7].

Procedure The procedure, approved by the local ethics committee, was in accordance with the relevant ethical standards of the Declaration of Helsinki (1964, revised in Fortaleza in 2013), and compliant with the EU GDPR. Participants were given an information sheet and asked to sign a consent form before the beginning of the experiment.

The experiments were conducted in part at a laboratory of the University of Trento and in part at the participants' home. In the latter case, participants were requested to conduct the experiment in a quiet and possibly acoustically isolated environment. Participants were instructed to adjust the volume of the sound to a reasonable level and keep such output level throughout the whole experiment. Subsequently, participants were asked to interact with the webMUSHRA application to listen to each trial and assign a score to it. The rankings were conducted over continuous faders ranging between 0 and 100. Each page featured 6 faders, one for each condition.

In this task, participants were asked to consider the following guiding question: *"If that was the quality of the incoming audio in an online musical interaction scenario - online lessons, rehearsal, jamming -, how much would you be disturbed?"* If they would be disturbed as much as from the reference track, then the score is 100; otherwise, the value should be found within the proposed rating scale.

Participants were exposed to the 9 audio-clips stimuli in the 6 conditions, for a total of 54 trials. Trials were

block-randomized by stimulus (i.e., the 6 conditions within each stimulus were randomized). Additionally, participants were instructed to take a 5-min break every 3 stimuli sessions.

No participants were excluded from the results analysis.

Results We perform an ANOVA on the results of the MUSHRA test using a linear mixed-effects model [35] having dataset and condition as fixed factors and participants as random factor. Post-hoc tests were performed on the fitted model using pairwise comparisons adjusted with the Tukey correction. The assumption on the normality of residuals was visually verified.

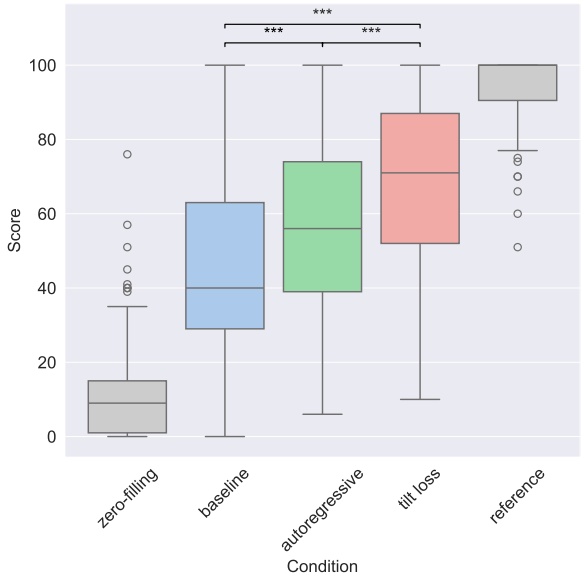


Fig. 4 Boxplot displaying the overall collected results from the user study. Legend: *** represents $p < 0.001$. Statistical significance is not displayed for pairs involving zero-filling and reference

A significant main effect was found for factors dataset ($F(2, 778) = 25.248$, $p < 0.001$) and condition ($F(5, 778) = 518.479$, $p < 0.001$), as well as for their interaction ($F(10, 778) = 12.453$, $p < 0.001$). In the following we report only the pairwise comparisons of interest for this study.

Regarding factor condition, overall Tilt Loss received significantly higher rankings than Baseline ($p < 0.001$) (see Fig. 4). This result indicates that the proposed method is generally able to outperform the baseline. Additionally, the autoregressive was also rated significantly higher than Baseline ($p < 0.001$). This is interesting, as an AR model of order 128 was used as baseline in the original PARCnet work. While the score of the AR method is still lower than the proposed methods, it is worth noting that using a higher-order model might yield much better results than the baseline method.

Nevertheless, grouping the response score per dataset (see Fig. 5) reveals that there are significant observations to be made on the models' performances in relation to each dataset. The boxplot of the Bach Cello Suite trials shows that the proposed model outperforms both the baseline and the improved AR model by a wide margin, having a mean score around 90. The statistical analysis confirms that the proposed method received higher rankings than Baseline and Autoregressive (both $p < 0.001$),

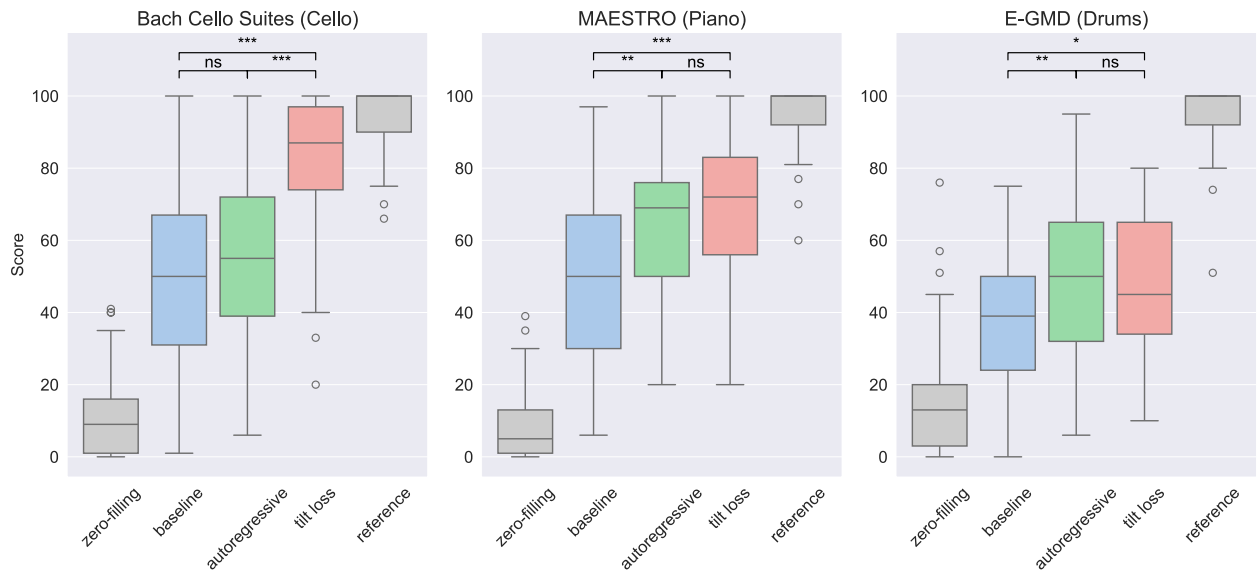


Fig. 5 Boxplot displaying the results by dataset. Legend: *** represents $p < 0.001$, ** represents $p < 0.01$, * represents $p < 0.05$ and *ns* represents $p > 0.05$. Statistical significance is not displayed for pairs involving zero-filling and reference

while no significant difference was found between Autoregressive and Baseline.

Conversely, the results on the MAESTRO dataset display a slightly less accentuated difference between the proposed method and the improved AR model, having around the same mean. The statistical analysis revealed that no significant difference was found between Tilt Loss and Autoregressive, while Baseline was rated significantly lower than Tilt Loss ($p < 0.001$ s) and Autoregressive ($p < 0.01$).

The result scores of the E-GMD trials are in slight contrast with the overall results as the improved AR model is roughly on par with the proposed method. No significant difference was found between Tilt Loss and Autoregressive, Tilt Loss and Baseline, and Baseline and Autoregressive. Additionally, they are significantly lower compared to all the other groupings. In fact, the mean of every concealment method is around 40, while in the other groupings, the proposed method was able to score a mean of up to 80.

This indicates that our method—and possibly the backbone PARCnet architecture itself—is less effective on percussive instruments, i.e., unpitched musical signals with transient behavior—high attack and high decay, low sustain and low release. Further research is required to better understand and achieve robust concealment of not linearly predictable signals.

5 Discussion

5.1 Objective vs subjective metrics in PLC

The absence of a reliable objective metric is a well-known problem in the field of PLC. The evaluation of PLC algorithms is a non-trivial task, as the traditional objective metrics used in the field of machine learning and signal processing are not able to capture the perceptual aspects of the audio signal, which are crucial in the context of PLC. With these limitations in mind, the evaluation of the proposed loss functions in this work was performed using a combination of objective and subjective metrics, with the underlying assumption that the latter would ensure the perceptual quality of the performances of the method.

The results of the user study confirmed that the proposed method outperforms the other baselines. Notably, these results are in plain contrast with the results of the objective metrics reported in Table 2, which show that the proposed method is outperformed by the other baselines. This is a clear indication that the current objective metrics do not provide any meaningful information in the evaluation of PLC algorithms.

These observations raise a critical concern on the reliability of the objective metrics used in the evaluation of PLC algorithms and suggest that more work is required from the community in developing novel metrics specifically tailored for the problem of PLC, which should be able to capture the perceptual aspects of musical audio and correlate with the evaluation of end users.

As a brief remark, the original PARCnet model was able to outperform all the other baselines not only in the perceptual evaluation of human subjects but also according to the several different objective metrics considered in that work ([7], III.A), hence featuring a correlation between the objective and subjective metrics. However, it is important to note that the other baselines considered for that work are either an ablation of the model itself—keeping only the AR branch, without modifications—or deep learning methods originally developed for speech tasks ([7], III). In conclusion, the correlation between objective and subjective metrics in the paper [7, III.B] does not constitute strong evidence of the reliability of the objective metrics in the evaluation of PLC algorithms.

5.2 Datasets and domain representation in PLC

In this work, we used three different datasets to provide a comprehensive evaluation of the proposed methods. However, such effort is not to be considered exhaustive and further research is needed to investigate the performance of the proposed methods on different datasets and domains. The domain of musical audio is vast and currently, there is a lack of a comprehensive dataset that can be used to train and evaluate the performance of PLC methods. In this work, the usage of three datasets aimed to maximize the coverage of the domain of musical audio, taking a step further from previous work in the literature, where methods were trained and evaluated on a single dataset, usually single instrument [6, 7] or small multi-instrument, like SQAM [4].

Machine learning and especially deep learning methods require massive amounts of high-quality data to function effectively, in the order of millions and possibly billions of datapoints. Additionally, a good dataset should be balanced enough according to some metrics in the domain and task of interest. The balance should hold both at the dataset level and at the split (train, validation, test) level. In this study, similarly to previous work, the only metric considered for the datasets is their size. Specifically, a subset of E-GMD and MAESTRO is taken so that they could easily match the size of the significantly smaller Bach Cello Suites; the size matching was performed at offline data preparation time and is determined by the PSF. Additionally, splits were performed at file level where, for each dataset, one file was held out as *validation set*, one as *test set* and all the remaining ones were used as training data.

In the data preparation pipeline used in this work, balance is not taken into account. Most notably, there is a

lack of consensus on what metrics should be used to evaluate a dataset in terms of *balance* for the task of PLC. For instance, in most classification tasks the dataset balance is evaluated in terms of class distribution; techniques such as oversampling, undersampling, or data augmentation are used to improve such aspects. However, in the case of PLC tasks, there is no standardized metric to evaluate such aspects.

This is a clear limitation of the current work, and it is suggested that future research should focus on this aspect. The results from the user study clearly indicate that the performance of the proposed method, as well as the baseline, varies significantly depending on the dataset, being more effective on pitched instruments than percussive instruments. While such a fact might be caused by the backbone model architecture's inability to encompass such type of audio signals, we argue that a high-quality dataset tailored for the task of PLC may help greatly in the development of models capable of higher quality concealments.

5.3 Considerations on real-time aspects

In this work, the PARCnet architecture and algorithm [7] were adopted without any modification, with the exception of the loss function used to train the model, which is the subject of this study. As loss functions are only employed in the offline training procedure, the considerations and evaluation of the real-time aspects of the model presented in the PARCnet work still hold. This is very convenient, as in the paper ([7], III.A, Table 2) it was reported that the PARCnet architecture was able to run an inference pass on a single packet on a consumer-grade CPU (AMD Ryzen 5900HS) without hardware acceleration in about 8.1 ms, which is within the time window of the packet, as packet length of 10 ms were considered. Having the model capable of real-time inference is a key aspect in the popularisation and widespread adoption of Networked Music Performance Systems, as it allows for the model to be deployed in actual application scenarios without expensive and power-hungry hardware.

As a final consideration, it is worth noting that the inference time reported from the paper was obtained running the model inference on Python and PyTorch by loading the Lightning checkpoint dump in a full instance of the PyTorch model class[2], which is known to be slower than other languages and frameworks [36] and hence is not a recommended method to deploy models in a production application. It is reasonable to expect that the inference time could be further reduced by porting the model to a more low-level language and platform, such as C++ and TensorFlow Lite. Moreover, the increased overhead may allow for deploying the algorithm with a better AR backbone, which may result in a better overall performance of the model.

[2] https://github.com/polimi-ispl/PARCnet/blob/main/parcnet/parcnet.py#L44.

6 Conclusions

In this work, an attempt was made to define a novel loss function for deep learning-based PLC algorithms that effectively correlates with the human evaluation of the concealed audio in the context of NMP. The results from the user study show that the proposed loss function is able to improve over the baseline according to the evaluation of human subjects for the Bach Cello Suites dataset. Conversely, the proposed method is outperformed by the baseline according to the objective metrics, which in turn demonstrates that they are inadequate to evaluate the results of this problem. Further analysis of the results shows that the performances of the proposed method, as well as other methods, vary depending on the instrument of the dataset. Specifically, performances are unsatisfactory on unpitched instruments, suggesting that more research effort should be put into understanding how to produce good concealments for non linearly predictable signals, possibly using deep learning-based PLC methods.

Future directions may involve refining the proposed loss function, such as developing a multi-resolution approach similar to that used in [7]. Furthermore, it would be worthwhile to investigate whether frequency reweighting provides advantages for architectures beyond PARCnet, particularly for non-hybrid methods that are entirely based on deep learning. Additionally, it would be convenient to define an objective metric tailored for the problem of PLC that correlates with the human evaluation. Having such a metric would provide industrial and academic researchers and engineers in PLC with a convenient mean to assess the performances of their algorithms, granting a major reduction in development time. A preliminary step towards this direction is represented by the recent study reported in [37]. In the context of deep learning-based PLC algorithms, another major future direction could involve the development of a comprehensive dataset tailored for training deep learning models. Such a dataset should encompass single-instrument recordings across various instruments as well as multi-instrument recordings. Additionally, it should ensure a robust representation of the musical audio domain, offering guarantees regarding its coverage and applicability.

Abbreviations

AR	Autoregressive
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DFT	Discrete Fourier Transform
E-GMD	Expanded Groove MIDI Dataset

FFT	Fast Fourier Transform
IoMusT	Internet of Musical Things
IoT	Internet of Things
MAE	Mean absolute error
MAESTRO	MIDI and Audio Edited for Synchronous TRacks and Organization
MOS	ITU. P.800.1 Mean Opinion Score
MSE	Mean squared error
MUSHRA	ITU. BS.1534 MUltiple Stimuli with Hidden Reference and Anchor
NMP	Networked Music Performance
NMPS	Networked Music Performance Systems
NMSE	Normalized mean squared error
NN	Neural network
PCM	Pulse Code Modulation
PEAQ	Perceptual Evaluation of Audio Quality
PESQ	Perceptual Evaluation of Speech Quality
PLC	Packet Loss Concealment
PSF	Packet skip factor
RAdam	Rectified Adam
STFT	Short-Time Fourier Transform
STOI	Short-Time Objective Intelligibility
UDP	User Datagram Protocol

Acknowledgements
The authors thank all the participants involved in the study.

Authors' contributions
FD conceptualized and implemented the proposed solution, designed and conducted the study, performed the analysis of the results, and did most of the writing. LV conceptualized the study and the proposed solution and provided technical support throughout the development, the analysis of the results and the writing. LT envisioned the research context and direction, performed the statistical analysis, and supervised and revised the writing. All authors read and approved the final manuscript.

Funding
The authors acknowledge the support of the MUSMET project funded by the EIC Pathfinder Open scheme of the European Commission (grant agreement n. 101184379). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council. Neither the European Union nor the European Innovation Council can be held responsible for them.

Data availability
The original code produced to train and evaluate the proposed solution is hosted on Github and is publicly accessible at https://github.com/CIMIL/tilt-loss. The datasets considered during the current study are available in the Bach Cello Suites Data Repository, https://ccrma.stanford.edu/~cc/som/bachCello, The MAESTRO Dataset Repository, https://magenta.tensorflow.org/datasets/maestro#v300, and The Expanded Groove MIDI Dataset Dataset Repository, https://magenta.tensorflow.org/datasets/e-gmd#v100.

Declarations

Competing interests
LT is a guest editor for the current special issue: Signal Processing for the Internet of Sounds. All other authors declare that they have no competing interests.

Received: 21 January 2025 Accepted: 13 December 2025
Published online: 09 January 2026

References

1. L. Turchet, C. Fischione, G. Essl, D. Keller, M. Barthet, Internet of musical things: vision and challenges. IEEE Access **6**, 61994–62017 (2018)
2. C. Rottondi, C. Chafe, C. Allocchio, A. Sarti, An overview on networked music performance technologies. IEEE Access **4**, 8823–8843 (2016)
3. C. Perkins, O. Hodson, V. Hardman, A survey of packet loss recovery techniques for streaming audio. IEEE Network **12**, 40–48 (1998)
4. M. Fink, Enhancements for networked music performances. phdthesis, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Hamburg, Germany. Organisational unit: Allgemeine Nachrichtentechnik. (2018). https://openhsu.ub.hsu-hh.de/handle/10.24405/4341. Accessed 26 Dec 2025
5. M. Sacchetto, Y. Huang, A. Bianco, C. Rottondi, in *Proceedings of the 17th International Audio Mostly Conference*. Using autoregressive models for real-time packet loss concealment in networked music performance applications, AM '22 (Association for Computing Machinery, New York, 2022), pp. 203–210
6. P. Verma, A.I. Mezza, C. Chafe, C. Rottondi, in *2020 27th Conference of open innovations association (FRUCT)*. A deep learning approach for low-latency packet loss concealment of audio signals in networked music performance applications (IEEE, Piscataway, NJ, USA, 2020), pp. 268–275
7. A.I. Mezza, M. Amerena, A. Bernardini, A. Sarti, Hybrid packet loss concealment for real-time networked music applications. IEEE Open J. Signal Process. **5**, 266–273 (2024)
8. A. Wright, V. Välimäki, in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Perceptual loss function for neural modeling of audio systems (2020), pp. 251–255. https://doi.org/10.1109/ICASSP40776.2020.9052944
9. F. Daniotti, L. Vignati, L. Turchet, Towards perceptual deep learning-based packet loss concealment. Technical report, University of Trento (2024). https://internetofsounds.net/public_downloads/Daniotti_tech_report.pdf. Accessed 26 Dec 2025
10. G. Zhang, W.B. Kleijn, in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, Autoregressive model-based speech packet-loss concealment (IEEE, 2008), pp. 4797–4800
11. J.D. Markel, A.J. Gray, *Linear prediction of speech*, vol. 12 (Springer Science & Business Media, New York, 2013)
12. P. Kabal, in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*. Ill-conditioning and bandwidth expansion in linear prediction of speech, vol. 1 (IEEE, Piscataway, NJ, USA, 2003), pp. I–I
13. S.O. Arık, H. Jun, G. Diamos, Fast spectrogram inversion using multi-head convolutional neural networks. IEEE Signal Process. Lett. **26**(1), 94–98 (2019)
14. C. Aironi, S. Cornell, L. Serafini, S. Squartini, in *2023 31st European Signal Processing Conference (EUSIPCO)*. A time-frequency generative adversarial based method for audio packet loss concealment (IEEE, Piscataway, NJ, USA, 2023), pp. 121–125
15. M. Sacchetto, C. Rottondi, A. Bianco, Implementation and optimization of burg's method for real-time packet loss concealment in networked music performance applications. Pers. Ubiquit. Comput. **28**, 1–17 (2024). https://doi.org/10.1007/s00779-024-01806-8
16. S. Butterworth, On the theory of filter amplifiers. Exp. Wirel. Wirel. Eng. **7**, 536–541 (1930)
17. C. Chafe. Bach cello suites data repository (2020). https://ccrma.stanford.edu/~cc/som/bachCello. Accessed 26 Dec 2025
18. C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.Z.A. Huang, S. Dieleman, E. Elsen, J. Engel, D. Eck, in *International Conference on Learning Representations*. Enabling factorized piano music modeling and generation with the MAESTRO dataset (2019). https://openreview.net/forum?id=r1lYRjC9F7. Accessed 26 Dec 2025
19. L. Callender, C. Hawthorne, J. Engel, Improving perceptual quality of drum transcription with the expanded groove midi dataset. CoRR abs/2004.00188 (2020). arXiv:2004.00188 [cs.SD].
20. V.A. Nguyen, A.H.T. Nguyen, A.W.H. Khong, in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Improving performance of real-time full-band blind packet-loss concealment with predictive network (IEEE, Piscataway, NJ, USA, 2023), pp. 1–5
21. J.M. Valin, J. Skoglund, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Lpcnet: Improving neural speech synthesis through linear prediction (IEEE, Piscataway, NJ, USA, 2019), pp. 5891–5895
22. J. Wang, Y. Guan, C. Zheng, R. Peng, X. Li, A temporal-spectral generative adversarial network based end-to-end packet loss concealment for wideband speech transmission. J. Acoust. Soc. Am. **150**, 2577–2588 (2021)

23. S. Pascual, J. Serrà and J. Pons, "Adversarial Auto-Encoding for Packet Loss Concealment," 2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), (New Paltz, 2021), pp. 71–75. https://doi.org/10.1109/WASPAA52581.2021.9632730
24. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E.Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems **32**, 8026–8037 (Neural Information Processing Systems Foundation, Inc., San Diego, CA, USA, Vancouver, Canada, 2019).
25. W. Falcon, The PyTorch Lightning team. PyTorch Lightning (2019). https://github.com/Lightning-AI/lightning. Accessed 26 Dec 2025
26. L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, in *International Conference on Learning Representations*, On the variance of the adaptive learning rate and beyond (2020). https://openreview.net/forum?id=rkgz2aEKDr. Accessed 26 Dec 2025
27. A. Rix, J. Beerends, M. Hollier, A. Hekstra, in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs (IEEE, Piscataway, NJ, USA, 2001), vol. 2. pp. 749–752
28. C.H. Taal, R.C. Hendriks, R. Heusdens, J. Jensen, in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. A short-time objective intelligibility measure for time-frequency weighted noisy speech (IEEE, Piscataway, NJ, USA, 2010), pp. 4214–4217
29. T.V. Thiede, W.C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J.G. Beerends, C. Colomes, Peaq - the ITU standard for objective measurement of perceived audio quality. J. Audio Eng. Soc. **48**, 3–29 (2000)
30. A.F. Khalifeh, A.K. Al-Tamimi, K.A. Darabkh, in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. Perceptual evaluation of audio quality under lossy networks (IEEE, Piscataway, NJ, USA, 2017), pp. 939–943
31. L. Diener, M. Purin, S. Sootla, A. Saabas, R. Aichner, R. Cutler, in *Proceedings of the 23rd Annual Conference of the International Speech Communication Association (INTERSPEECH)*. Plcmos – a data-driven non-intrusive metric for the evaluation of packet loss concealment algorithms (International Speech Communication Association (ISCA), Incheon, 2022)
32. ITU. P.800.1 : Mean opinion score (MOS) terminology. https://itu.int/rec/T-REC-P.800.1. Accessed 26 Dec 2025
33. ITU. BS.1534 : Method for the subjective assessment of intermediate quality level of audio systems. https://www.itu.int/rec/R-REC-BS.1534/en. Accessed 26 Dec 2025
34. M. Schoeffler, S. Bartoschek, F.R. Stöter, M. Roess, S. Westphal, B. Edler, J. Herre, Webmushra — a comprehensive framework for web-based listening tests. J. Open Res. Softw. **6**(1), 1–8 (2018). https://doi.org/10.5334/jors.187
35. D. Bates, M. Mächler, B. Bolker, S. Walker, Fitting linear mixed-effects models using lme4. J. Stat. Softw. **67**(1), 1–48 (2015)
36. D. Stefani, S. Peroni, L. Turchet, in *Proceedings of the 25-th Int. Conf. on Digital Audio Effects (DAFx20in22)*. A Comparison of Deep Learning Inference Engines for Embedded Real-Time Audio Classification, vol. 3 (DAFx, Vienna, Austria, 2022), pp. 256–263
37. L. Vignati, L. Turchet, On the lack of a perceptually-motivated evaluation metric for packet loss concealment in networked music performances. J. Audio Eng. Soc. **73**(10), 660–670 (AES, New York, 2025). https://doi.org/10.17743/jaes.2022.0227

Publisher's Note