Is Self-Supervised Learning Enough to Fill in the Gap? A Study on Speech Inpainting

Ihab Asaad, Maxime Jacquelin, Olivier Perrotin, Laurent Girin, Thomas Hueber

Is Self-Supervised Learning Enough to Fill in the Gap? A Study on Speech Inpainting

Ihab Asaad[a,b,1], Maxime Jacquelin[a,c,1], Olivier Perrotin[a], Laurent Girin[a], Thomas Hueber[a,*]

[a] *Univ. Grenoble Alpes, CNRS, Grenoble-INP, GIPSA-lab, Grenoble, F-38000, France*
[b] *Friedrich Schiller Universität Jena, Jena, Germany*
[c] *Vogo, Bernin, 38190, France*

Abstract

Speech inpainting consists in reconstructing corrupted or missing speech segments using surrounding context, a process that closely resembles the pretext tasks in Self-Supervised Learning (SSL) for speech encoders. This study investigates using SSL-trained speech encoders for inpainting without any additional training beyond the initial pretext task, and simply adding a decoder to generate a waveform. We compare this approach to supervised fine-tuning of speech encoders for a downstream task—here, inpainting. Practically, we integrate HuBERT as the SSL encoder and HiFi-GAN as the decoder in two configurations: (1) fine-tuning the decoder to align with the frozen pre-trained encoder's output and (2) fine-tuning the encoder for an inpainting task based on a frozen decoder's input. Evaluations are conducted under single- and multi-speaker conditions using in-domain datasets and out-of-domain datasets (including unseen speakers, diverse speaking styles, and noise). Both informed and blind inpainting scenarios are considered, where the position of the corrupted segment is either known or unknown. The proposed SSL-based methods are benchmarked against several baselines, including a text-informed method combining automatic speech recognition with zero-shot text-to-speech synthesis. Performance is assessed using objective metrics and perceptual evaluations. The results demonstrate that both approaches outperform baselines, successfully reconstructing speech segments up to 200 ms, and sometimes up to 400 ms. Notably, fine-tuning the SSL encoder achieves more accurate speech reconstruction in single-speaker settings, while a pre-trained encoder proves more effective for multi-speaker scenarios. This demonstrates that an SSL pretext task can transfer to speech inpainting, enabling successful speech reconstruction with a pre-trained encoder.

Keywords: Speech inpainting, self-supervised model, speech synthesis, speech enhancement, neural vocoder.

1. Introduction

In speech processing, inpainting aims at restoring a speech signal that has been "locally" and severely corrupted. This means that a segment of the waveform is either highly degraded (e.g., by clicks or clipping (Záviška et al., 2021)), or missing (e.g., due to packet loss during transmission (Perkins et al., 1998; Thirunavukkarasu and Karthikeyan, 2015)). This type of degradation severely affects and can even completely alter the signal quality and intelligibility for the corrupted segment and its direct neighbourhood, while the rest of the signal remains intact. This contrasts with speech enhancement in noise (Loizou, 2007), where a more or less stationary additive noise is present throughout the entire signal duration. Speech inpainting has been explored in the past few decades using a variety of techniques (see Section 2 for a detailed review of the literature). Initially, classical signal processing was applied to short corrupted segments (Gunduzhan and Momtahan, 2001; Lagrange et al., 2005). This was followed by the use of deep neural networks (DNNs)-based encoder-decoder architectures, which directly map incomplete signals to the complete ones, through fully-supervised training (Kegler et al., 2020; Zhao, 2023). In particular, the encoder processes the signal surrounding the corrupted segment, while the decoder generates the signal within the corrupted segment. Lastly, speech inpainting have been considered as one "downstream" task of a model pre-trained with Self-Supervised Learning (SSL), i.e., by fine-tuning the model to the inpainting task (Xue et al., 2023; Zhang et al., 2024a).

Interestingly, speech inpainting shares notable similarities with SSL that uses masking as a pretext task. Transformer-based SSL encoders such as HuBERT (Hsu et al., 2021), wav2vec (Schneider et al., 2019), wav2vec2.0 (Baevski et al., 2020), and WavLM (Chen et al., 2022), learn rich representations by masking portions of the input, and by training the model to predict the representations of the masked input segments using information from the unmasked context. While this process does not occur directly in the signal domain but in a non-linearly projected embedding space, both speech

*Corresponding author
[1] Ihab Asaad and Maxime Jacquelin contributed equally to this work.

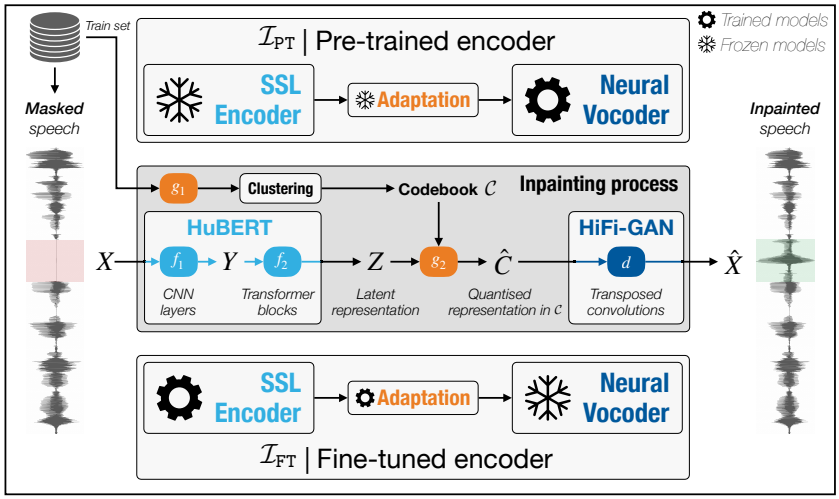


Figure 1: Proposed inpainting framework, leveraging a pre-trained SSL encoder on a unmasking task and using a neural vocoder for audio generation. Top: the SSL encoder is kept frozen (pre-trained) while the neural vocoder is adapted to the SSL output. Bottom: the SSL encoder is fine-tuned to an inpainting task, while the neural vocoder is kept frozen. In the present study we use HuBERT as the SSL and HiFi-GAN as the vocoder (subfigure in the middle). The inpainting process and adaptation mechanism between the SSL output and neural vocoder input are detailed in Section 3).

inpainting and SSL share the objective of leveraging regularities across multiple time scales and linguistic levels to infer the content of either a severely corrupted, missing or masked segment within an input speech signal. Having drawn this parallel, to the best of our knowledge, the SSL masking pretext task has never been evaluated as such in the context of speech inpainting. *We thus hypothesise that the mask prediction task used in SSL can be directly applied to speech inpainting, without the need for fine-tuning the SSL encoder.*

To test this hypothesis, we investigate in this study the integration of an SSL encoder—specifically, HuBERT (Hsu et al., 2021)—into a speech inpainting pipeline, utilising its rich contextual representations, spanning from phonetics to semantics (Pasad et al., 2023), to "fill in the gap" by reconstructing the severely corrupted or missing portions of a speech signal based on its surrounding context. More specifically, we utilise the HuBERT model for inference, maintaining the same configuration as in its SSL pre-training phase. We input signals that contain missing segments, and the model predicts the unmasked segments, which we then use to fill in the missing portions of the signals. Since HuBERT does not directly predict the missing time-domain signal samples but rather high-dimensional embeddings, we thus combine the SSL encoder with a neural vocoder—in the present case HiFi-GAN (Kong et al., 2020). It is important to stress out that compared to (Xue et al., 2023) and (Zhang et al., 2024a), we do not fine-tune the SSL encoder. We separately train the neural vocoder to learn the mapping between the output of the pre-trained SSL model and the time-domain speech signal, as directly inspired from (Polyak et al., 2021). We denote this inpainting method $\mathcal{I}_{\text{PT}}$ (with PT standing for pre-trained), which is summarised in the top of Fig. 1.

For comparison, we then assess whether fine-tuning the SSL model to the inpainting task, i.e. treating inpainting as a downstream task instead of a pretext SSL training task, further enhances performance. We denote this inpainting method $\mathcal{I}_{\text{FT}}$ (with FT standing for fine-tuned), as shown at the bottom of Fig. 1. In this case, we fine-tune the SSL encoder to fill in missing portions of the input signal directly in the audio domain. We choose a Mel-spectrogram representation for this purpose, as it can maintain the same temporal resolution as the original HuBERT representation space and serves as the standard input of a vanilla HiFi-GAN to convert back to the time-domain. Overall, let us note that our two approaches are symmetrical: $\mathcal{I}_{\text{PT}}$ involves a frozen pre-trained encoder with an adapted decoder, while $\mathcal{I}_{\text{FT}}$ involves a fine-tuned encoder with a frozen decoder.

Lastly, we compare the ability of an SSL encoder (specifically, HuBERT) for inpainting against a generative model. We introduce a baseline inspired by the text-informed speech inpainting study of Prablanc et al. (2016). Their approach combines text-to-speech synthesis (TTS) and voice conversion (VC) to inpaint the corrupted portion of the audio signal, using the text from both uncorrupted and corrupted portions of the signal. We adapted their method to a textless inpainting scenario by incorporating an ASR frontend based on the Whisper model (Radford et al., 2023) to decode the most likely word sequence from the signal to be inpainted and replacing the TTS-followed-by-VC pipeline by a more recent zero-shot neural TTS. This baseline represents an additional contribution of this work.

In this paper, we do not aim to propose a novel method for speech inpainting that outperforms state-of-the-art methods. Instead, we investigate whether the straightforward but so far unexplored solution of using a pre-trained SSL model as-is for speech inpainting is viable. To this end, we assess the performance of the proposed SSL-based methods in both single-

speaker and multi-speaker settings, using in-domain datasets, which aligned with the training data, and out-of-domain datasets, which feature unseen speakers, varying speaking styles (including expressive speech), and different types of background noise—challenging conditions that, to the best of our knowledge, have received limited attention in previous studies. Performance is evaluated with both objective metrics and perceptual tests. To sum up, the contributions of this paper are:

1. An extensive state-of-the-art of inpainting methods;
2. The direct application of the mask prediction task used in SSL to speech inpainting (inpainting as a pretext task — $\mathcal{I}_{\mathrm{PT}}$);
3. The comparison with fine-tuning of SSL model to inpainting (inpainting as a downstream task — $\mathcal{I}_{\mathrm{FT}}$);
4. An extensive evaluation of both methods against different baselines and on in-domain and out-of-domain datasets;
5. The complete source code[2] and a demo page for the two proposed speech inpainting frameworks and the different baselines.[3]

2. Related work

2.1. Speech inpainting

Early signal processing approaches. Speech inpainting studies initially focused on short corrupted segments (i.e., from a few milliseconds to a few tens milliseconds), using conventional digital signal processing approaches. The first targeted applications were packet loss recovery in telecommunications and streaming audio (Perkins et al., 1998; Thirunavukkarasu and Karthikeyan, 2015) or signal declicking (Záviška et al., 2021). Early works were based on signal processing techniques such as linear prediction / auto-regressive models (Etter, 1996; Gunduzhan and Momtahan, 2001; Kondo and Nakagawa, 2006; Zhang and Kleijn, 2008), Hidden Markov models (Rodbro et al., 2006; Borgström et al., 2010), sinusoidal modelling (Lindblom and Hedelin, 2002; Lagrange et al., 2005), or graphs (Perraudin et al., 2018).

Deep learning based approaches. As already stated in the introduction, the speech inpainting problem has been tackled more recently with deep learning approaches. With the huge development of Voice on IP (VoIP) technologies, packet loss cancellation (PLC) has been a driving force application for deep learning-based speech inpainting studies. The domain has rapidly evolved in a few years considering longer corrupted segments (consecutive short-term frames), going from a few tens of milliseconds to several hundreds of milliseconds and more, and denser distributions of loss packets, with packet loss rates ranging from a few percent to more than 50 %, possibly simulating a high level of VoIP degradation.

In terms of architectures and training methodology, earlier studies have considered direct incomplete-to-complete signal mapping with a single network and fully-supervised training, most often using a few tens of hours of speech signal. This was applied in the time-frequency (TF) domain, using, e.g., a basic fully-connected neural network (FCNN) (Lee and Chang, 2016), a U-Net architecture (Kegler et al., 2020; Dai et al., 2024), a combination of U-Net with temporal convolutional network (TCN) bottleneck (Aironi et al., 2025), or a convolutional recurrent network (CRN) (Zhang et al., 2024b). Other works have considered using a pair of 1D convolutional neural network (CNN) encoder and decoder for waveform analysis and reconstruction, e.g., (Shi et al., 2019; Wang et al., 2021; Zhu et al., 2022; Liu et al., 2022). Finally, some studies have considered working directly on the time-domain waveform samples using a recurrent neural network (RNN), e.g., (Lotfidereshgi and Gournay, 2018; Mohamed and Schuller, 2020; Westhausen and Meyer, 2022), or a CRN (Lin et al., 2021). Many works have considered the generative adversarial network (GAN) training approach (Goodfellow et al., 2014), i.e., combining a generative network with a discriminative network at training time, for improving the quality of the inpainted speech signal, e.g., (Shi et al., 2019; Wang et al., 2021; Pascual et al., 2021; Liu et al., 2022; Zhao, 2023; Aironi et al., 2025; Dai et al., 2024; Zhang et al., 2024b).

More complex encoder-decoder combinations have been proposed over the years. For example, a hybrid STFT-domain encoder combined with a time-domain CNN decoder was used in (Pascual et al., 2021). Taking inspiration from deep learning-based speech synthesis, and in particular the recent developments in text-to-speech synthesis using neural vocoders, speech inpainting and PLC studies have considered combining a missing features predictor network with a (possibly pre-trained) neural vocoder. For example, a GRU-based RNN designed to predict missing LPCNet input features was combined with an LPCNet vocoder in (Valin et al., 2022). A relatively basic FCNN-based Mel-spectrogram predictor taking as input the 11 previous frames (i.e., 120 ms context) was combined with a WaveGlow vocoder in (Zhou et al., 2022). A full-band recurrent network (FRN) combining an encoder, a predictor and a joiner network was proposed in (Nguyen et al., 2023b). This FRN model was improved in (Yang and Chang, 2023) by adding a CTC-based ASR representation loss in the training, and in (Yang et al., 2023) by combining it with a masked frequency prediction for bandwidth extension. Recent works have integrated a Transformer module (Vaswani et al., 2017) within the

[2] `https://gricad-gitlab.univ-grenoble-alpes.fr/huebert/speech-inpainting`
[3] `http://www.ultraspeech.com/demo/csl_2025_inpainting/`

inpainting model, to exploit its natural ability to predict a (missing) sequence portion from a large temporal context. For example, the combination of CNN and Transformer was proposed in (Li et al., 2023; Zhao et al., 2024) and a GAN TCN with a Conformer predictive network was proposed in (Zhao et al., 2023). Li et al. (2022) introduced a wav2vec-based perceptual loss term in the GAN training of their 1D CNN encoder-decoder model (in addition to the reconstruction loss and an ASR-based loss term). Very recently, a Flow Matching approach was proposed in (Yang and Chang, 2025). The two PLC challenges (Diener et al., 2022, 2025) summarise and compare the most recent inpainting methods for packet loss concealment. They have demonstrated excellent performance from the best systems in terms of intelligibility and naturalness of the resulting speech.

Finally, closely related to our work, Xue et al. (2023) and Zhang et al. (2024a) introduced an SSL representation at the encoder level and combined it with a neural decoder. Xue et al. (2023) implement an encoder with two parallel branches: one acoustic branch, made of 2D convolution and self-attention blocks, and one wav2vec2.0-like (Baevski et al., 2020) semantic branch pre-trained with contrastive learning. The outputs of these two branches are merged and send to an HiFi-GAN vocoder (Kong et al., 2020). Importantly, Xue et al. (2023) train the semantic branch from scratch using packet loss as the masking scheme, before training end-to-end the rest of the model with the same data. Zhang et al. (2024a) propose a model similar to that of Xue et al. (2023), though with an additional temporal-dilated module to fuse the wav2vec2.0-like semantic branch and the acoustic branch before decoding the output waveform with an in-house decoder. Overall, both studies use SSL models as "feature extractors" and perform end-to-end fine-tuning of their network on inpainting tasks. Conversely, we hypothesise that a pre-trained SSL encoder is inherently capable of performing an inpainting task, providing it with a decoder for domain adaptation.

Multimodal speech inpainting. A few studies have investigated the use of a visual input such as the speaker's lips for guiding the speech inpainting process. This was implemented with an LSTM- or Transformer-based multimodal context encoder (Morrone et al., 2021; Hsu et al., 2023). Other speech inpainting studies have investigated text-informed methods, where the complete text (or phonetic information, or sequence of linguistic targets) is provided, covering both the uncorrupted and corrupted parts of the signal, and used to drive the inpainting process (Prablanc et al., 2016; Borsos et al., 2022). Text-informed speech restoration was also considered in (Koizumi et al., 2023). In this study, we do not consider any additional modality, we assume that only the locally-corrupted speech signal is available, and we thus focus on audio-only speech inpainting.

2.2. Speech enhancement in noise, speech restoration, and speech coding

As briefly stated in the introduction, speech enhancement in noise is a task that is related to speech inpainting in the sense that the speech signal is degraded and must be restored, but the nature of the degradation is significantly different. Therefore, the restoration process is also expected to be significantly different. However, after decades of development of speech enhancement methods working in the TF domain and using both conventional signal processing techniques (Loizou, 2007) and deep learning (Wang and Chen, 2018), recent studies in speech enhancement have proposed to use a combination of neural feature extractor with predictor taking the noisy speech as input and a neural vocoder for clean speech (re)generation, e.g., (Maiti and Mandel, 2019; Du et al., 2020; Liu et al., 2021; Su et al., 2021; Saeki et al., 2022; Irvin et al., 2023; Koizumi et al., 2023). This approach is closely similar to some of the above-mentioned inpainting studies, and brings to the speech enhancement problem the advantage of avoiding the delicate problem of signal phase reconstruction when only the magnitude spectrogram is denoised. We can note that HiFi-GAN is the preferred vocoder in (Saeki et al., 2022; Irvin et al., 2023). We can also note that in fact, the noisy feature extractor in (Irvin et al., 2023; Koizumi et al., 2023) is an SSL model, making these studies closely related to our work in terms of model architecture and combination (HuBERT with HiFi-GAN is even among the multiple combinations tested by Irvin et al. (2023)). However, both studies focus on speech enhancement in noise and dereverberation, and do not address inpainting specifically. Moreover, the model in (Koizumi et al., 2023) also takes text as input, whereas our study focuses on audio-only inpainting.

A few recent papers have reconsidered the speech enhancement problem as a generalized multi-task problem, possibly including denoising, dereverberation, bandwidth extension, declipping and packet loss concealment (hence inpainting) under the general denomination of speech restoration, see, e.g., (Liu et al., 2021; Saeki et al., 2022; Koizumi et al., 2023). However, in practice, in their experiments, these studies do not consider the inpainting task as defined in the present study, that is the recovery of long highly-corrupted or missing segments of speech. For example, Saeki et al. (2022) introduced their work in the context of speech restoration but only consider equalization and bandwidth extension in the experiments.

Finally, it can be noted that specifically combining HuBERT and HiFi-GAN was also proposed by Polyak et al. (2021) for speech coding, where the input signal is unaltered, and the primary objective is to generate an output signal that closely matches the input, while drastically limiting the bitrate of the encoded speech representation. As stated in the introduction, we inspired from this study but focus on speech inpainting, i.e., using the unmasking abilities of the SSL encoder in inference, which is a very different task than speech coding.

2.3. Speech continuation with speech language models

Speech continuation shares the task of predicting a missing portion of a speech signal with speech inpainting but is causal (prediction from a past context only), and operates over portion lengths that are orders of magnitude longer than those of conventional speech inpainting applications such as PLC. As a consequence, the most recent approaches leverage a generative speech language model (SpeechLM) for this task, i.e., a model trained to generate speech via auto-regressive prediction and decoding of quantised elementary speech units referred to as speech "tokens" (see (Arora et al., 2025) for a recent review). Lakhotia et al. (2021) have demonstrated the ability of such models for audio-only speech continuation. Borsos et al. (2023) and Chen et al. (2025) have further combined the prediction of audio tokens with "semantic" tokens that are derived from text-based language models. In addition to semantic tokens, Du et al. (2024) used ground-truth text as input for the missing speech portion to predict.

Overall, while those methods could be considered as suitable candidates for inpainting tasks, they do not fit the main hypothesis of this paper. These methods leverage the generative ability of auto-regressive models for the speech continuation task while we take a step aside in studying the parallel between the unmasking pretext task of SSL encoders and speech inpainting. Moreover, the SSL-based pipeline investigated here is significantly lighter than recent SpeechLM approaches, which is essential for most practical use cases (e.g. embedded voice prothesis, real-time PLC). Nevertheless, as mentioned in introduction, we use a generative method as a baseline for comparison. We choose Whisper (Radford et al., 2023) which is based on a generative text-based language model, as text tokens are currently more efficient than audio tokens for next token prediction and therefore constitutes a stronger baseline than audio token-based methods.

2.4. Conclusion

In conclusion, SSL are becoming ubiquitous in a large variety of speech generation tasks, including speech coding, speech restoration, speech enhancement, speech continuation and speech inpainting. Nevertheless, to the best of our knowledge, none of the related studies have considered an SSL encoder pre-trained on an unmasking task as a suitable candidate for speech inpainting without the need for fine-tuning. In the following Sections, we therefore present our implementation of a pre-trained HuBERT model in an inpainting pipeline, as well as its evaluation.

3. Method

3.1. Problem formulation

Following the notations introduced in (Mohamed et al., 2022), let $X = \{x_1, \ldots, x_T\}$ denote a sequence of speech samples of length T (i.e., a waveform), and let $X_{-[t_1,t_2]}$ denote the sequence where the segment $X_{t\in[t_1,t_2]} = \{x_{t_1}, x_{t_1+1}, \ldots, x_{t_2}\}$ is corrupted (e.g., degraded or missing), and that we wish to reconstruct. In this paper, we focus on *non-causal* inpainting, where the inpainting function $\mathcal{I}$ has access to both past and future uncorrupted portions of the input signal. We consider both the *informed* inpainting paradigm, in which the corrupted segment position (i.e., the pair of values $\{t_1, t_2\}$) is known, and the *blind* paradigm, in which the corrupted segment position is unknown. In the informed configuration, only the corrupted speech segment is predicted from its surrounding context. The uncorrupted portions of the signal are kept as is and are combined with the reconstructed segment using cross-fade at junctions to mitigate artifacts commonly associated with the concatenation of raw speech or audio segments. This process is illustrated in the right part of Fig. 2c. It is formalised as:[4]

$$\hat{X}_{t\in[t_1,t_2]} = \mathcal{I}\left(X_{-[t_1,t_2]}\right) \quad \text{and} \quad \hat{X}_{t\notin[t_1,t_2]} = X_{t\notin[t_1,t_2]}. \tag{1}$$

In blind inpainting, the entire output signal is generated without distinguishing between the corrupted and uncorrupted parts, as illustrated in the right part of Fig. 2d. This can be simply formalized as:

$$\hat{X} = \mathcal{I}\left(X_{-[t_1,t_2]}\right). \tag{2}$$

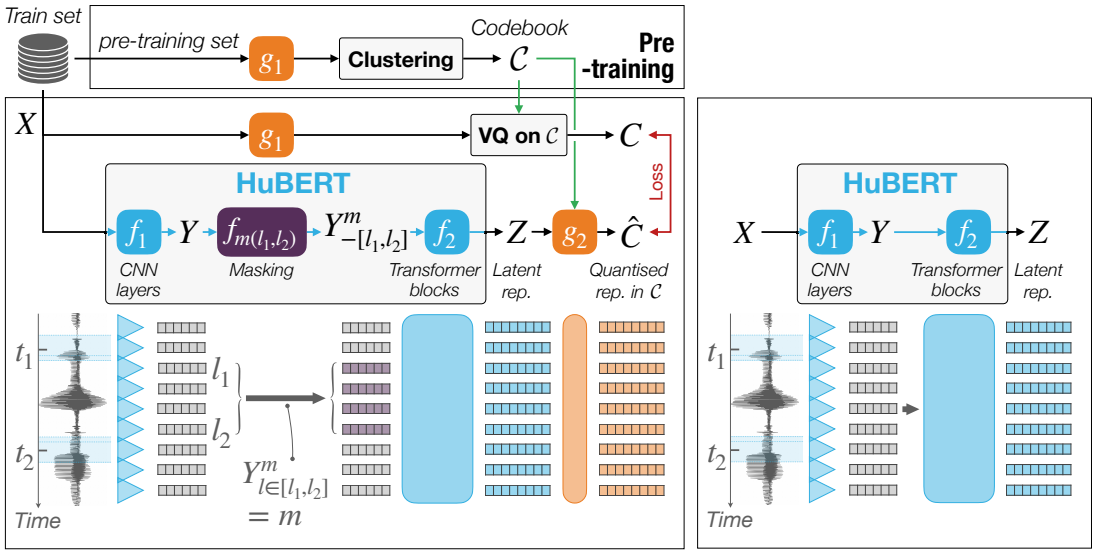
3.2. Masking, the core pretext task of HuBERT

In this subsection, we briefly present the principles and core equations of HuBERT, as this formalism will be essential for explaining our inpainting pipeline in the subsequent subsections. HuBERT is an encoder that transforms a speech signal X of length T into a latent continuous representation $Z = \{z_1, \ldots, z_L\}$, with dimension $D_z \times L$ (Hsu et al., 2021):

$$y_l = f_1\left(X_{[lH-u,lH+u[}\right), \forall l \in \{1, L\}, \tag{3}$$

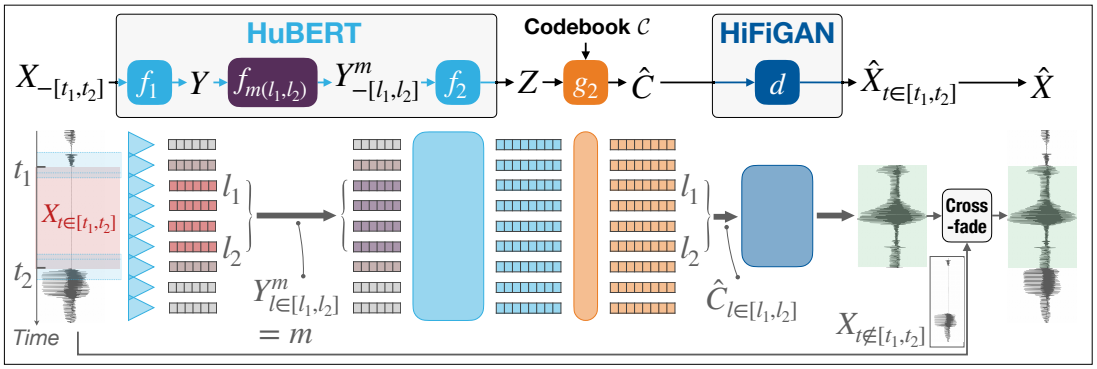
$$Z = f_2\left(Y\right), \quad \text{with} \quad Y = \{y_1, \ldots, y_L\}, \tag{4}$$

[4]For simplicity of presentation, we omit the cross-fade operation in this equation.

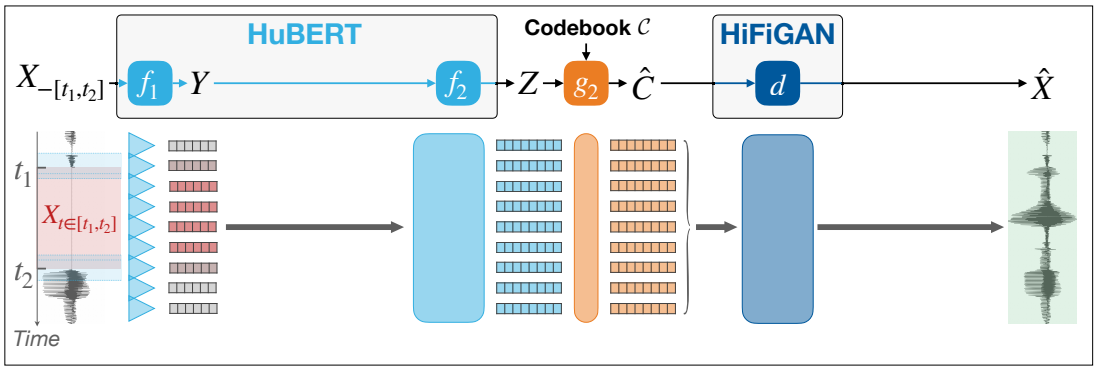


(a) HuBERT training

(b) HuBERT inference



(c) HuBERT-based inpainting pipeline – informed case (inference)



(d) HuBERT-based inpainting pipeline – blind case (inference)

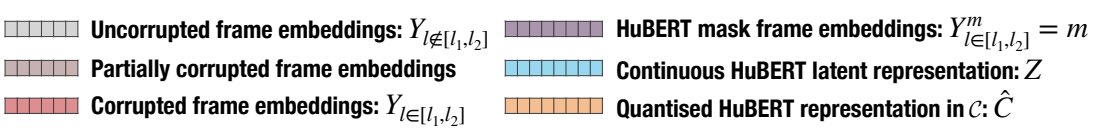


Figure 2: Illustration of how the HuBERT model works during training (top-left) and inference (top-right). Adaptation of HuBERT for inference within the proposed inpainting pipeline, in the informed (middle) and blind (bottom) inpainting configurations.

where f_1, sometimes referred to as the prenet, is a stack of CNNs of span $2u$ samples and hop size H, and f_2 is a stack of Transformer encoder blocks. These core modules of HuBERT are illustrated in Fig. 2b for inference. We do not detail their architecture in this paper, the reader is referred to the original HuBERT (Hsu et al., 2021) and Transformer papers (Vaswani et al., 2017) for details.

HuBERT is trained iteratively to predict a vector-quantised representation of the speech signal, denoted as C, from the latent representation Z. For instance, during the first training iteration of the vanilla HuBERT, C is a sequence of quantised Mel-frequency cepstral coefficient (MFCC) vectors. This is done with the help of two auxiliary modules, g_1 and g_2, which function as teacher and student models, respectively (see Fig. 2a):

- The teacher module g_1 maps the speech signal to the new representation (unquantised MFCC vectors in the above example), with a span of $2w$ samples and window shift H. Before training, a codebook C of quantised prototype vectors is created by passing a portion of the training set through g_1 and applying a k-means algorithm on the output. Then, during training, g_1 and vector quantisation (VQ) are used to extract a target quantised sequence $C = \{c_1, \ldots, c_L\}$ from each waveform of the train dataset:

$$c_l \quad = \quad \mathtt{VQ}_C\big(g_1(X_{[lH-w,lH+w[})\big), \forall l \in \{1, L\}, \tag{5}$$

  where $\mathtt{VQ}_C$ stands for VQ using the codebook C.

- The student module g_2 is designed to predict the quantised sequence $\hat{C}$ from the encoder output Z. In practice, g_2 is implemented using a softmax function, which involves a (learned) linear projection of z_l onto embedding vectors representing the codebook prototypes.

Importantly, during training, a portion of the CNN-encoded sequence Y is randomly masked before being used to generate the fully-encoded sequence Z. Specifically, a sub-sequence $Y_{l\in[l_1,l_2]}$ of Y, randomly selected, is replaced with $(l_2 - l_1 + 1$ occurrences of) a mask embedding vector m, which is jointly learned during the training process. This masking operation, spanning frames l_1 to l_2, is illustrated in Fig. 2a. We denote it as:

$$Y^m_{-[l_1,l_2]} = f_{m(l_1,l_2)}(Y). \tag{6}$$

In summary, during HuBERT training, we have:

$$y_l = f_1\left(X_{[lH-u,lH+u[}\right), \qquad \forall l \in \{1, L\}, \tag{7}$$

$$Z = f_2 \circ f_{m(l_1,l_2)}\left(Y\right), \qquad \text{with } Y = \{y_1, \ldots, y_L\} \text{ and } Z = \{z_1, \ldots, z_L\}, \tag{8}$$

$$\hat{c}_l = g_2\left(z_l\right), \qquad \forall l \in \{1, L\}. \tag{9}$$

During HuBERT training, g_1 is fixed, while f_1, f_2 and g_2 are updated to minimise the distance between $\hat{C}$ and C, with part of the Y sequence randomly masked across batches. Note that g_1 and g_2 are only used for HuBERT training and are generally discarded during inference when HuBERT is applied to a downstream task. Similarly, in conventional HuBERT usage, masking is only applied during training (see Fig. 2a and (8)) and it is discarded during inference (see Fig. 2b and (4)). Consequently, the pre-trained HuBERT consists solely of $f_2 \circ f_1$, as defined in (3) and (4), and a newly trained module dedicated to the downstream task is typically appended to f_2.

3.3. Linking masking, prediction and inpainting

The proposed HuBERT-based inpainting pipeline relies on the following general principle. As discussed in the previous subsection, HuBERT is trained to predict an efficient speech representation Z from a Y sequence containing one or several masked segments. The pretext task used for training HuBERT shares similarities with inpainting in its underlying principles. Indeed, in both cases, the goal is to reconstruct an output sequence from an input sequence that includes missing, corrupted, or masked portions. The difference lies in the nature of the input and output: (i) During HuBERT training, the reconstructed/predicted output is the speech representation Z, whereas in inpainting, it is directly the signal X; and (ii) during HuBERT training, the missing input information corresponds to the masked part of the CNN-encoded sequence Y, whereas in inpainting, it is a segment of the input signal X. In summary, HuBERT is trained to perform a form of speech inpainting, but from the Y space to the Z space, and not directly within the input signal space. Consequently, we integrate HuBERT in a speech inpainting pipeline, addressing points (i) and (ii) as follows.

To address point (i), and drawing inspiration from recent advancements in low-bit-rate neural speech codecs (Polyak et al., 2021), we combine HuBERT as an encoder with the neural vocoder HiFi-GAN (Kong et al., 2020) as a decoder. HiFi-GAN, used here to reconstruct the inpainted speech waveform $\hat{X}$ from HuBERT's output Z, demonstrated state-of-the-art performance in the latest text-to-speech synthesis benchmark (Perrotin et al., 2025). Its necessary adaptation to the proposed inpainting pipeline is detailed in the next subsection.

For point (ii), we remark that, since the prenet temporal span is fixed, the corrupted input signal segment $X_{t\in[t_1,t_2]}$ we aim to inpaint corresponds to a corrupted subset of vectors $Y_{l\in[l_1,l_2]}$ in the CNN-encoded sequence Y. Therefore, inpainting the corrupted signal $X_{-[t_1,t_2]}$ relies on HuBERT's ability to predict Z accurately from the corresponding complete (corrupted) sequence $Y_{-[l_1,l_2]}$. However, leaving the corrupted subsequence $Y_{l\in[l_1,l_2]}$ as is in Y is suboptimal because, during HuBERT training, the vectors composing this subsequence are replaced with the mask m (see Section 3.2). For HuBERT, this results in a mismatch between training (on masked segments) and inference (on corrupted segments, i.e., its use in the inpainting pipeline) conditions. To resolve this issue, we explicitly replace $Y_{l\in[l_1,l_2]}$ with the mask m *during inference*. In other words, we apply (8) instead of (4) at inference/inpainting time. This approach is feasible, however, only in the informed case, where the time sample boundaries (t_1, t_2) of the corrupted input speech segment are known, allowing us to deduce the corresponding frame boundaries (l_1, l_2) (see the left part of Fig. 2c). In the blind case, the location of $Y_{l\in[l_1,l_2]}$ is unknown and we cannot apply the mask (see the left part of Fig. 2d). In this case, we thus stick to (4) at inference/inpainting time and have to accept that HuBERT functions under a train/inference mismatch. The blind inpainting configuration can be seen as a way to evaluate HuBERT's ability to infer an accurate speech representation from the surrounding context of the missing/corrupted speech information, even when this missing/corrupted speech information is not explicitly indicated by a mask embedding vector.

Note that, as mentioned at the end of Section 3.2, in the various uses of HuBERT reported in the literature, masking is applied only during training and is never used during inference. This is because, when using the speech representation Z in downstream tasks, the input signal X is generally *not* corrupted. To the best of our knowledge, the speech inpainting pipeline considered in this paper represents the first instance where the masking process, which is central to HuBERT's training pretext task, is explicitly used *at inference time* (in the informed configuration).

3.4. Combining HuBERT encoder with HiFi-GAN decoder

In this subsection, we present in detail how we combined the HuBERT encoder with the HiFi-GAN decoder. To this end, we first recall that during training on masked Y sequences, HuBERT predicts Z via a quantised representation (see (9)). Therefore, to carefully match the pretext training task during inpainting, we retain the auxiliary module g_2, which transforms Z into the quantised sequence $\hat{C}$. $\hat{C}$ is then used as input to the HiFi-GAN decoder (denoted d) (see Figs. 1 and 2). In the informed case, we only use the subsequence $\hat{C}_{l\in[l_1,l_2]}$ (corresponding to the masked portion of Y) to reconstruct the missing speech segment. The latter is then combined with the uncorrupted parts of the input signal (see Fig. 2c). This can be expressed as:[5]

$$\begin{cases} \hat{X}_{t\in[t_1,t_2]} &= d(\hat{C}_{l\in[l_1,l_2]}) \\ \hat{X}_{t\notin[t_1,t_2]} &= X_{t\notin[t_1,t_2]}, \end{cases} \tag{10}$$

where again, $[l_1, l_2]$ is the frame interval corresponding to the corrupted segment of input signal sample interval $[t_1, t_2]$. In the blind case, we simply have (see Fig. 2d):

$$\hat{X} = d(\hat{C}). \tag{11}$$

In the following, we detail how to adapt g_2 to interface HuBERT with HiFi-GAN, and how to accordingly configure g_1 and the codebook C to train g_2. This adaptation depends on the two *frameworks* presented in introduction: (i) using a pre-trained (PT) HuBERT, where the HiFi-GAN decoder is trained to fit a frozen HuBERT encoder, and (ii) fine-tuning the SSL encoder (FT), where we fine-tune the HuBERT encoder to the inpainting task, with an output that fits the standard input of a frozen HiFi-GAN decoder. These methods are illustrated in Fig. 1 and detailed below.

3.4.1. Pre-trained SSL

In this first approach, we use a pre-trained HuBERT model[6] and keep it frozen. To adapt the HiFi-GAN decoder to the frozen pre-trained HuBERT, we adopt the following two-step adaptation process. In the first step, we directly use Z as the new signal representation (i.e., $g_1^{(\mathrm{PT})}$ is identical to the frozen pre-trained HuBERT $f_2 \circ f_1$). A new codebook $C^{(\mathrm{PT})}$ is obtained by running the k-means algorithm on the Z sequences derived from a subset of the pre-training dataset. $g_2^{(\mathrm{PT})}$ is then simply the quantisation on $C^{(\mathrm{PT})}$ of the encoded sequence Z corresponding to any corrupted input sequence $X_{-[t_1,t_2]}$:

$$\hat{C} = g_2^{(\mathrm{PT})}(Z) = \mathtt{VQ}_{C^{(\mathrm{PT})}}(Z). \tag{12}$$

In the second step, we trained a HiFi-GAN model $d^{(\mathrm{PT})}$ from scratch to reconstruct the speech waveform from HuBERT's output. Specifically, the decoder takes the index of each $\hat{c}_l$ in the codebook $C^{(\mathrm{PT})}$ as input and learns a look-up table of embedding vectors, which are then fed into the standard HiFi-GAN architecture.

[5] Again, for simplicity of presentation, we omit the cross-fade operation in this equation.

[6] The term "pre-trained" here refers to both the SSL training with masking and the ASR fine-tuning, as discussed in Section 3.4.2.

By denoting with * the modules that are trained, the full inpainting pipeline $\mathcal{I}_{\mathrm{PT}}$ can be summarized as follows. In the informed case, we have:

$$\begin{aligned}\hat{X}_{t\in[t_1,t_2]} &= \mathcal{I}_{\mathrm{PT}}\left(X_{-[t_1,t_2]}\right)\\ &= d^{(\mathrm{PT})*}\circ\left\{g_2^{(\mathrm{PT})}\circ f_2\circ f_{m(l_1,l_2)}\circ f_1(X_{-[t_1,t_2]})\right\}_{l\in[l_1,l_2]},\end{aligned} \tag{13}$$

and in the blind case, we have:

$$\begin{aligned}\hat{X} &= \mathcal{I}_{\mathrm{PT}}\left(X_{-[t_1,t_2]}\right)\\ &= d^{(\mathrm{PT})*}\circ g_2^{(\mathrm{PT})}\circ f_2\circ f_1\left(X_{-[t_1,t_2]}\right).\end{aligned} \tag{14}$$

3.4.2. Fine-tuned SSL

In this second approach, we use the standard (vanilla) HiFi-GAN decoder, which takes a Mel-spectrogram as input and remains frozen, while we adapt HuBERT. To fully understand our adaptation of HuBERT and the motivation behind it, we must revisit the conventional HuBERT training process, briefly outlined in Section 3.2.

As described in details in (Hsu et al., 2021), after HuBERT is initially pre-trained on a masking pretext task using g_1 and g_2, it is subsequently fine-tuned on an ASR task. In this stage, g_1 and g_2 are discarded, and f_2 is extended with a softmax layer for connectionist temporal classification of characters, and fine-tuned using a labelled speech dataset. In our FT approach to inpainting, we instead aim at predicting signal frames to fill-in the gap, while aligning with HiFi-GAN's Mel-spectrogram input representation. To achieve this, we reintroduce g_1 and g_2 and perform a new round of training using the masking pretext task on a reasonable amount of data. In this iteration, the speech representation is replaced by the Mel-spectrogram. Altogether, this adaptation process can be seen an inpainting-oriented fine-tuning. Conceptually, it can be seen as partially "unlearning" ASR-specific abstractions in order to recover more detailed signal properties. Importantly, however, the model still benefits from HuBERT's strong ability to encode the contextual structure surrounding the gap.

In a few more details, we define $g_1^{(\mathrm{FT})}$ as the extraction of Mel-spectrogram vectors from (frames of $2w$ samples of) the waveforms X. The codebook $C^{(\mathrm{FT})}$ is obtained with the k-means algorithm applied on the output of $g_1^{(\mathrm{FT})}$ for a train set. Given an input sequence X, the teacher module $g_1^{(\mathrm{FT})}$ computes a Mel-spectrogram vector for each frame, which is assigned to its closest centroid in $C^{(\mathrm{FT})}$ (as in (5)). The student softmax-based $g_2^{(\mathrm{FT})}$ module of (Hsu et al., 2021) is also re-introduced, to predict a sequence $\hat{C}$ of quantised Mel-spectrogram vectors in $C^{(\mathrm{FT})}$:

$$\hat{c}_l = g_2^{(\mathrm{FT})}(z_l) = \underset{c}{\operatorname{argmax}}\left(\frac{\exp\left(\operatorname{sim}\left(Az_l, e_c^{(\mathrm{FT})}\right)/\tau\right)}{\sum_{c'=1}^{\operatorname{Card}(C^{(\mathrm{FT})})}\exp\left(\operatorname{sim}\left(Az_l, e_{c'}^{(\mathrm{FT})}\right)/\tau\right)}\right), \tag{15}$$

where A is a linear projection, $\operatorname{sim}(a,b)$ is the cosine similarity between a and b, $e_c^{(\mathrm{FT})}$ is a learnt embedding of codeword $c\in C^{(\mathrm{FT})}$, and τ is the logit scale factor (Hsu et al., 2021), set to 0.1. Following HuBERT training described in Section 3.2, $g_1^{(\mathrm{FT})}$ and f_1 are fixed, and f_2 and $g_2^{(\mathrm{FT})}$ are updated to minimise the cross-entropy loss between the predicted indices of Mel-spectrogram vectors $\hat{c}_l$ (softmax output of (15)) and the indices of the quantised Mel-spectrogram vectors $c_l\in C^{(\mathrm{FT})}$, while part of the Y sequence is randomly masked across batches. At inference, the sequence $\hat{C}$ of quantised Mel-spectrogram vectors is directly fed to a pre-trained vanilla HiFi-GAN.

By noting again by * the modules that are trained, the full $\mathcal{I}_{\mathrm{FT}}$ framework can be summarized as follows. In the informed case, we have:

$$\begin{aligned}\hat{X}_{t\in[t_1,t_2]} &= \mathcal{I}_{\mathrm{FT}}\left(X_{-[t_1,t_2]}\right)\\ &= d\circ\left\{g_2^{(\mathrm{FT})*}\circ f_2^*\circ f_{m(l_1,l_2)}\circ f_1(X_{-[t_1,t_2]})\right\}_{l\in[l_1,l_2]},\end{aligned} \tag{16}$$

and in the blind case, we have:

$$\begin{aligned}\hat{X} &= \mathcal{I}_{\mathrm{FT}}\left(X_{-[t_1,t_2]}\right)\\ &= d\circ g_2^{(\mathrm{FT})*}\circ f_2^*\circ f_1\left(X_{-[t_1,t_2]}\right).\end{aligned} \tag{17}$$

4. Experimental set-up

4.1. Datasets

In-domain train/test sets. We conducted experiments on both LJ Speech (Ito and Johnson, 2017) and VCTK (Yamagishi et al., 2019) datasets. LJ Speech is an English corpus containing 13 100 short audio clips recorded by a single female

speaker for a total length of approximately 24 h. We isolated 12 950 clips as the training/validation set, the remaining 150 clips being used for test. VCTK includes a set of 43 859 audio clips recorded by 109 English speakers balanced in gender and with various accents, for a total of approximately 44 h. We used 41 747 clips from 105 speakers for training and 389 clips from 4 speakers for test. Importantly, we carefully designed the partitioning of the VCTK dataset to have no overlap between the training and test sets in terms of sentences and speakers. In other words, the proposed models have to generalise to both new speakers and new linguistic content.

Out-of-domain test sets. We also considered two supplementary datasets for evaluating the extrapolation capabilities of our models to out-of-domain data. First, we designed noisy versions of our LJ Speech and VCTK test sets with the same number of utterances as described before (i.e., 150 for LJ Speech and 389 for VCTK). On each clip, we then applied either a white noise or a crowd noise with three signal-to-noise ratio (SNR) levels (20, 10 and 0 dB) for a total of six noise-corrupted signals per utterance. Those two new test sets account for a total of 900 utterances for LJ Speech and 2334 for VCTK.

Second, we used a subset of the Expresso dataset (Nguyen et al., 2023a), a multi-speaker expressive speech dataset covering seven different speaking styles uttered by four different speakers. Following the recommendation from the authors of the Expresso dataset, our test set includes 588 utterances, with 147 clips per speaker and 84 clips per speaking style.

4.2. Implementation details

SSL encoder HuBERT. For the two proposed inpainting frameworks $\mathcal{I}_{\mathrm{PT}}$ and $\mathcal{I}_{\mathrm{FT}}$, we used the HuBERT-large model *hubert-large-ls960-ft*, publicly available on Hugging Face (see our repository). This model is a fine-tuned version of *hubert-large-ll60k*, the latter being initially trained on the Libri-Light dataset (Kahn et al., 2020), including 60 000 h of speech data from over 7000 speakers. The fine-tuning was done on the LibriSpeech dataset, containing 960 h of speech data from over 2484 speakers. To the best of our knowledge, the LJ Speech and VCTK datasets used in the present study are not included in LibriSpeech. However, since LJ Speech is extracted from the LibriVox[7] dataset, there might be a slight overlap with the very large Libri-Light dataset (also based on LibriVox). However, this overlap is approximately 0.0004% (24 vs. 60 000 hours) and thus remains very limited.

In the HuBERT model used in this study, the speech input is expected to be sampled at 16 kHz. The prenet window size and hop size are 400 and 320 samples (i.e., 25 and 20 ms), respectively. The output Z has dimension 768.

Inpainting with pre-trained SSL ($\mathcal{I}_{\mathrm{PT}}$). For this approach, we used the implementation of the speech encoder-decoder framework proposed by Polyak et al. (2021). Dedicated codebooks $C^{(\mathrm{PT})}$ were computed using the LJ Speech (resp. VCTK) dataset, considering a training subset of 21 h (resp. 36 h). For the single-speaker setting, we choose 100 clusters, similarly to the first pre-training iteration of HuBERT (i.e., for k-means clustering of MFCC features) (Hsu et al., 2021). For the multi-speaker setting, we found in preliminary experiments that increasing the codebook size to 500 did provide noticeable qualitative improvements, while offering a good trade-off between performance and computational cost. We therefore adopt this size. Recall that for this model $g_2^{(\mathrm{PT})}$ is not trained, i.e., for any representation Z of a masked input sequence $X_{-[t_1,t_2]}$, $g_2^{(\mathrm{PT})}$ retrieves the closest sequence of vectors $\hat{C} = \{\hat{c}_1, \ldots, \hat{c}_L\}$ from $C^{(\mathrm{PT})}$. HiFi-GAN is then trained from scratch to generate $\hat{X}$ from $\hat{C}$. Note that contrary to (Polyak et al., 2021), we did *not* use a fundamental frequency (f_o) encoder as an input in parallel to the $\hat{C}$ sequence in order to avoid the explicit prediction of the f_o of the corrupted segment. For the multi-speaker configuration (i.e., model trained on the VCTK dataset), a speaker embedding extracted using the speaker identification model proposed in (Heigold et al., 2016) was used as an additional conditioning vector to HiFi-GAN. Here, we trained this model on the same VCTK training subset than for the codebook computation (36 h). Both the HiFi-GAN vocoder and the speaker identification model were trained with the Adam optimiser (Kingma and Ba, 2015) over 200 epochs, with a batch size of 32 and a learning rate of 2×10^{-4}.

Inpainting with fine-tuned SSL ($\mathcal{I}_{\mathrm{FT}}$). In this second approach, the HuBERT encoder is fine-tuned to directly predict a quantised Mel-spectrogram representation, which is converted back to a time-domain audio signal with HiFi-GAN. More precisely, $g_2^{(\mathrm{FT})}$ is an additional linear layer designed to adapt the HuBERT output Z to a quantised Mel-spectrogram domain, as defined in (15). The Transformer blocks f_2 are fine-tuned, while $g_2^{(\mathrm{FT})}$ is trained from scratch. For this sake, the teacher $g_1^{(\mathrm{FT})}$ computes 80-dimensional Mel-spectrogram vectors from the input speech, with a window size of 46 ms and a hop size of 20 ms. As for the $\mathcal{I}_{\mathrm{PT}}$ framework, dedicated codebooks $C^{(\mathrm{FT})}$ were computed on the LJ Speech (resp. VCTK) training subset. We used 100 (resp. 500) clusters, but with the k-means applied on Mel-spectrogram vectors obtained from $g_1^{(\mathrm{FT})}$. For each training set (LJ speech or VCTK), fine-tuning f_2 and training $g_2^{(\mathrm{FT})}$ was done using the Adam optimiser over 100 epochs, with a batch size of 8 and a learning rate of 10^{-4}.

[7] `https://librivox.org`

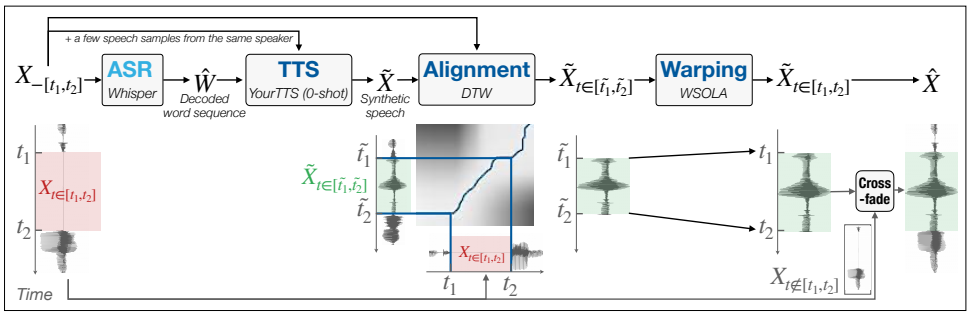


Figure 3: ASR-TTS baseline $\mathcal{I}_{\texttt{ASR}}$ combining automatic speech recognition (ASR) and zero-shot text-to-speech (TTS) for text-informed speech inpainting. The corresponding inpainting process is detailed in Section 4.3).

For the audio synthesis, we used a pre-trained HiFi-GAN model,[8] taking an 80-dimensional Mel-spectrogram as input, and generating a waveform at 22.05 kHz.[9] While being optional, a slight fine-tuning of HiFi-GAN on quantised ground-truth Mel-spectrograms was found to be beneficial for the overall audio quality. This was done using Adam over 50 epochs, with a batch size of 8 and a learning rate of 10^{-4}.

Post-processing. For the blind inpainting case, the reconstructed signal was generated entirely by the neural vocoder. For the informed case, we only kept the generated signal corresponding to the masked part and we placed it within the original (masked) signal using a cross-fade of 5 ms on both sides. Finally, the inpainted signals obtained with the $\mathcal{I}_{\texttt{FT}}$ framework were resampled to 16 kHz for a fair comparison with the other framework and baselines.

4.3. Baselines

Linear baseline. As a first baseline, we implemented a simple inpainting method based on linear interpolation ($\mathcal{I}_{\texttt{LI}}$). For a given masked signal, it consists in calculating its Mel-spectrogram (as done in Section 4.2) and replacing the masked frames with a linear interpolation between the last frame before the mask and the first frame after the mask. The interpolated Mel-spectrogram was then fed to the pre-trained HiFi-GAN vocoder to generate a 22.05 kHz waveform, which was then downsampled to 16 kHz.

Comparison with other studies. We also wanted to compare the proposed speech inpainting frameworks with other recently published methods such as (Kegler et al., 2020; Zhao, 2023; Xue et al., 2023) and (Zhang et al., 2024a). Unfortunately, no source code is publicly available for these studies and we could not perform experiments with these methods with the exact same configurations as the ones used in the proposed frameworks. Nevertheless, since we use common metrics (detailed below), in Section 5 we compare our results with the ones reported in (Kegler et al., 2020; Zhao, 2023; Xue et al., 2023), and (Zhang et al., 2024a) at least in terms of order of magnitude.

ASR-TTS baseline. In addition to the linear baseline and the results reported in comparable studies, we introduced an additional baseline based on generative modelling and inspired by text-informed speech inpainting, particularly the study by Prablanc et al. (2016). For this study, we adapted their method to a text-less inpainting scenario by incorporating an automatic speech recognition (ASR) frontend to decode the most likely word sequence from the signal to be inpainted. This adaptation leverages the ASR's internal language model to recover the linguistic content of the corrupted portion using its surrounding context. The resulting approach, denoted as $\mathcal{I}_{\texttt{ASR}}$, is presented in Fig. 3 and can be summarized as follows:

1. ASR is used to decode the most likely word sequence, $\hat{W}$, from the signal to be inpainted, $X_{-[t_1,t_2]}$.[10]
2. Zero-shot TTS (also known as voice cloning) is used to synthesize $\tilde{X}$ using both $\hat{W}$ and the signal $X_{-[t_1,t_2]}$. The zero-shot TTS is preferred to the TTS-followed-by-VC approach used in (Prablanc et al., 2016) since it allows to generate a synthetic speech signal close to the target speaker's timbre $\tilde{X}$ in a single processing step.[11]
3. The synthetic signal $\tilde{X}$ is time-aligned with the original signal $X_{-[t_1,t_2]}$ using Dynamic Time Warping (DTW). This alignment identifies which segment of the synthetic signal, $\tilde{X}_{t\in[\tilde{t}_1,\tilde{t}_2]}$, corresponds most closely to the missing chunk in the signal to inpaint, $X_{t\in[t_1,t_2]}$.

[8] More specifically, we used the `UNIVERSAL_V1` model (see our repository).

[9] An upsampling of HuBERT's codebook, initially computed considering 16 kHz speech input, was therefore necessary.

[10] We used the state-of-the-art *whisper-large* model available on *HuggingFace*; `https://huggingface.co/openai/whisper-large`

[11] We used the *YourTTS* system (Casanova et al., 2022), as implemented in the *coqui-TTS* toolkit; `https://github.com/coqui-ai/TTS`. Depending on the dataset used, if $X_{-[t_1,t_2]}$ was too short to allow the TTS speaker encoder to effectively capture the speaker's timbre, additional utterances from the same speaker were concatenated to provide approximately 10 seconds of speech.

4. The identified synthetic chunk is stretched or compressed to match the target duration $(t_2 - t_1)$, replacing its original duration $(\tilde{t_2} - \tilde{t_1})$. This adjustment is performed using the Waveform Similarity-based Overlap-Add (WSOLA) algorithm (Verhelst and Roelands, 1993).
5. The adjusted synthetic segment is inserted into $X_{-[t_1,t_2]}$ with cross-fade to ensure a seamless transition.

Importantly, this approach is applicable only to the informed inpainting paradigm, as it requires knowledge of the exact boundaries of the portion to be inpainted (t_1 and t_2). It is also worth noting that this approach is relatively straightforward, as it does not involve any model training or fine-tuning; both the ASR and zero-shot TTS components are pre-trained and used "off-the-shelf."

4.4. Evaluation metrics

Objective metrics. We evaluated the inpainted speech quality using PESQ (Rix et al., 2001) and its intelligibility using STOI (Taal et al., 2010) on both LJ Speech and VCTK test sets, comprising 150 and 389 utterances for the non-noisy versions and 900 and 2334 utterances for the noisy versions. To further evaluate the zero-shot learning capabilities of our models, we conducted the same experiments on the Expresso test set, consisting of 588 utterances. In all datasets, each utterance was masked three times using mask lengths of 100, 200 and 400 ms, and whose position was randomly chosen in each utterance, resulting in a total of 4361 $\times$ 3 masked utterances to inpaint with our four frameworks ($\mathcal{I}_{\texttt{LI}}$, $\mathcal{I}_{\texttt{ASR}}$, $\mathcal{I}_{\texttt{PT}}$, $\mathcal{I}_{\texttt{FT}}$). PESQ and STOI were computed by considering segments of one second of speech, centred on the mask (therefore, the inpainted speech corresponds to 10, 20, or 40 % of the original speech when measuring the score). As a complementary objective evaluation, we also performed automatic speech recognition (ASR) on the inpainted speech. We used the same pre-trained Whisper model (Radford et al., 2023) as in our baseline $\mathcal{I}_{\texttt{ASR}}$ and report the character error rate (CER).[12] For all metrics, average scores on each test set and each mask length are reported for all systems, with the binomial proportion confidence interval.

Perceptual evaluation. To further investigate the performance of the proposed inpainting system, we conducted an online MUSHRA-based listening test using the Web Audio Evaluation Tool (Jillings et al., 2015). This was only done for the informed case. First, we randomly sampled 15 sentences from the LJ Speech test set (single-speaker condition), and 15 sentences from the VCTK test set (multi-speaker condition). For each sentence, we randomly masked a 200 ms-long segment and inpainted it with $\mathcal{I}_{\texttt{FT}}$, $\mathcal{I}_{\texttt{PT}}$, and with the $\mathcal{I}_{\texttt{LI}}$ and $\mathcal{I}_{\texttt{ASR}}$ baselines. Finally, we asked 100 native English speakers (self-reported as British or American, recruited via the Prolific platform[13]) to evaluate the quality of the inpainted speech using a MUSHRA-based protocol (ITU, 2015). For each sentence, presented in random order, participants had to rate comparatively the four inpainted signals as well as a high-anchor signal (natural uncorrupted speech). The type of each signal (natural or inpainted) was not given to participants. As a reference, participants also received both the original uncorrupted sound file and its textual transcription. For each signal, they were instructed to focus on the inpainted speech segment which was highlighted using square brackets around the corresponding textual transcription, and asked to rate whether the highlighted speech segment was similar to that of the reference, on a continuous scale. Following the post-screening procedure described in (ITU, 2015), we excluded 48 participants who ranked the hidden natural speech less than 90 out of 100, for more than 15% of the stimuli (resulting in a final set of 52 participants who were considered as performing the test correctly). Participants took a median time of 19 min to complete the test and were compensated 3.75 £.

Statistical analysis. In the following, we assess the effect of the *mask length* (100, 200, 400 ms), *framework* ($\mathcal{I}_{\texttt{LI}}$, $\mathcal{I}_{\texttt{ASR}}$, $\mathcal{I}_{\texttt{PT}}$, $\mathcal{I}_{\texttt{FT}}$), *dataset* (single- vs. multi-speaker) and *type of inpainting* (informed vs. blind) factors, when relevant, on the objective and subjective metrics. The *utterance* was added as a random factor. We each time use a beta regression model (using the R function `glmmTMB`), followed by post-hoc pairwise comparisons between factor levels (R function `glht`). Details of factors involved in each statistical analysis are given in the next Section. Significance level is systematically set to $p < 0.01$.

5. Results

5.1. Qualitative results

Examples of inpainted speech signals obtained with the two proposed frameworks ($\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{PT}}$) and with the baselines $\mathcal{I}_{\texttt{LI}}$ and $\mathcal{I}_{\texttt{ASR}}$, in the informed case, and for a mask length of 200 ms, are presented in Fig. 4. Other examples, for other mask lengths and datasets, are available on our demo webpage[14]. We first examine the spectral pattern observed for

[12] This metric provides useful information about the phonetic content of the inpainted speech but may be biased by the linguistic prior on which the ASR may rely to transcribe it.

[13] `https://www.prolific.co`

[14] `http://www.ultraspeech.com/demo/csl_2025_inpainting/`

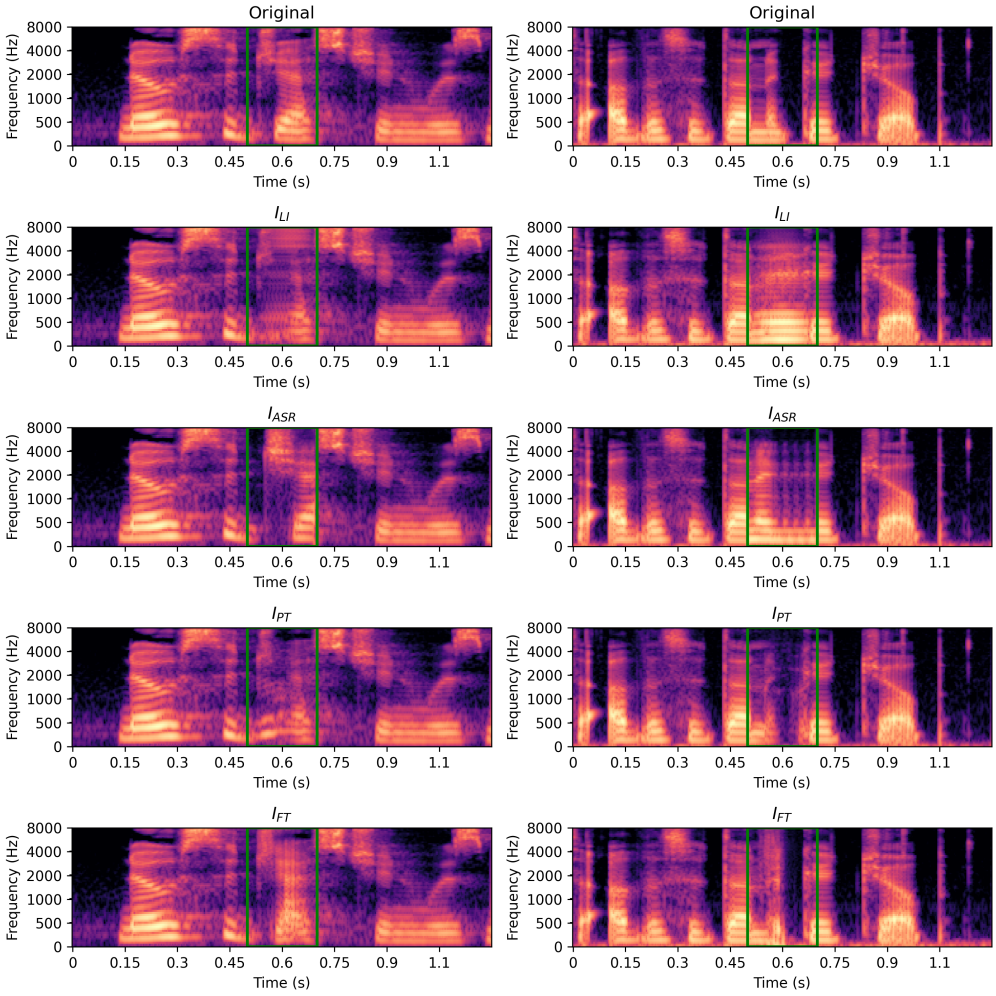


Figure 4: Examples of inpainted speech signals (80-dimensional Mel-spectrograms, informed case). Left: single-speaker for the sentence “no su[gge]stion was made", Right: multi-speaker for the sentence “(...) has do[ne a goo]d job". The green rectangles illustrate the position and length of the mask (200 ms).

the linear baseline $\mathcal{I}_{\texttt{LI}}$. Recall that the Mel-spectrogram is computed from the audio output of the HiFi-GAN vocoder, the latter being fed with a linearly interpolated mel-spectrogram between the beginning and the end of the mask. Interestingly, despite this “linear input”, the inpainted speech is almost —but not entirely— stationary. In fact, for the single-speaker case (left column), we can observe a transient a few milliseconds after the start of the mask. Consequently, the neural vocoder has “shaped” the linear input (likely not seen in its training corpus), probably by exploiting contextual information. However, as our quantitative evaluation confirms (see Section 5.2.1), this minimal sound shaping is not precise enough to recover the phonetic content of the masked part and the speech inpainted by the $\mathcal{I}_{\texttt{LI}}$ framework is most often not intelligible.

We now qualitatively compare the two SSL-based frameworks $\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{PT}}$ and the $\mathcal{I}_{\texttt{ASR}}$ baseline. For the single-speaker case (left column), the signal to be reconstructed corresponds to approximately two phones: a post-alveolar affricate /ʧ/ followed by a vowel /ɛ/ (in the word *suggestion*). The complex spectral pattern associated with this phonetic sequence is better reconstructed by $\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{ASR}}$ frameworks than with $\mathcal{I}_{\texttt{PT}}$, with a sharper vowel-consonant transition ($\mathcal{I}_{\texttt{PT}}$ wrongly maintains a strong formants structure during the consonant). For the multi-speaker case (right column), the signal to inpaint corresponds to the phonetic sequence [n ə g ʊ]. Here, $\mathcal{I}_{\texttt{FT}}$ is less efficient. It correctly reconstructs the initial nasal n as well as the plosive g and the final vowel ʊ but surprisingly replaces the middle schwa ə with an unvoiced and high energy sound, creating a kind of audio artefact. This is not the case with the $\mathcal{I}_{\texttt{PT}}$ framework, with which the signal is very well reconstructed. Regarding $\mathcal{I}_{\texttt{ASR}}$, the phonetic content is well reconstructed, as the linguistic targets are accurately decoded by the ASR. However, the transition between ə and g sounds less natural than with $\mathcal{I}_{\texttt{PT}}$, likely due to the absence of a clear closure before g, as indicated by the sound energy just before the burst of g. These initial qualitative results are confirmed by the quantitative evaluation presented in the following sections.

Table 1: Informed speech inpainting results. For all metrics, average scores with confidence intervals for each test set, each mask length, and each framework, for both the single- and multi-speaker configurations. The best scores per framework are denoted in bold. Pairs of symbols indicate pairs of distributions that are not significantly different.

		Single-speaker (LJ Speech)			**Multi-speaker (VCTK)**		
Models	Mask (ms)	PESQ $[-0.5; 4.5] \uparrow$	STOI $[0; 1] \uparrow$	CER (%) $\downarrow$	PESQ $[-0.5; 4.5] \uparrow$	STOI $[0; 1] \uparrow$	CER (%) $\downarrow$
Unmasked	0	4.25 ± 0.01	0.97 ± 0.00	6 ± 1	4.05 ± 0.01	0.94 ± 0.01	4 ± 1
Baseline $\mathcal{I}_{\mathrm{LI}}$	100	2.92 ± 0.04 ■	0.92 ± 0.01	13 ± 2 ●	2.40 ± 0.02	0.87 ± 0.01	17 ± 1
	200	2.25 ± 0.04	0.83 ± 0.01	16 ± 2 ◆	1.97 ± 0.02	0.77 ± 0.01	13 ± 1 ▲
	400	1.95 ± 0.03	0.71 ± 0.01	19 ± 1 ★	1.57 ± 0.02	0.60 ± 0.01	22 ± 1 ▼
Baseline $\mathcal{I}_{\mathrm{ASR}}$	100	2.97 ± 0.05 ■	0.93 ± 0.01 ● ●	13 ± 1 ●	2.78 ± 0.03	0.90 ± 0.01	11 ± 1 ▲
	200	2.63 ± 0.06	0.88 ± 0.02 ● ◆	17 ± 1 ◆	2.46 ± 0.03	0.85 ± 0.01 ◀	13 ± 1 ▼ ▲
	400	1.86 ± 0.05	0.82 ± 0.02	21 ± 1 ★ ■	1.93 ± 0.03	0.79 ± 0.01 ▶	24 ± 1 ▼
$\mathcal{I}_{\mathrm{PT}}$	100	3.06 ± 0.04	0.94 ± 0.01 ●	15 ± 1 ■	$\mathbf{3.13 \pm 0.01}$	$\mathbf{0.93 \pm 0.01}$	$\mathbf{8 \pm 1}$
	200	2.85 ± 0.04	0.89 ± 0.01 ▶	13 ± 2 ◆ ■ ▼	$\mathbf{2.93 \pm 0.01}$	$\mathbf{0.88 \pm 0.01}$ ▶	15 ± 1 ▼
	400	2.78 ± 0.04	$\mathbf{0.86 \pm 0.01}$ ★	24 ± 3 ■	$\mathbf{2.66 \pm 0.02}$	$\mathbf{0.83 \pm 0.01}$	$\mathbf{18 \pm 1}$ ●
$\mathcal{I}_{\mathrm{FT}}$	100	$\mathbf{3.28 \pm 0.01}$	$\mathbf{0.96 \pm 0.01}$	$\mathbf{7 \pm 1}$	3.06 ± 0.01	0.90 ± 0.01	10 ± 1 ▲
	200	$\mathbf{3.09 \pm 0.02}$	$\mathbf{0.93 \pm 0.01}$ ◆	$\mathbf{12 \pm 1}$ ◆ ▲	2.70 ± 0.02	0.85 ± 0.01 ◀	$\mathbf{12 \pm 1}$ ▼ ▲
	400	$\mathbf{2.93 \pm 0.03}$	$\mathbf{0.86 \pm 0.01}$ ★	$\mathbf{14 \pm 1}$	2.39 ± 0.02	0.79 ± 0.02 ▶	19 ± 1 ●

Table 2: Blind speech inpainting results. For all metrics, average scores with confidence intervals for each test set, each mask length, and each framework, for both the single- and multi-speaker configurations. The best scores per framework are denoted in bold. Pairs of symbols indicate pairs of distributions that are not significantly different.

		Single-speaker (LJ Speech)			**Multi-speaker (VCTK)**		
Models	Mask (ms)	PESQ $[-0.5; 4.5] \uparrow$	STOI $[0; 1] \uparrow$	CER (%) $\downarrow$	PESQ $[-0.5; 4.5] \uparrow$	STOI $[0; 1] \uparrow$	CER (%) $\downarrow$
$\mathcal{I}_{\mathrm{PT}}$	0	2.87 ± 0.02	0.89 ± 0.01	19 ± 2	3.11 ± 0.01	0.93 ± 0.01	13 ± 1
	100	2.77 ± 0.04	0.88 ± 0.01 ▼	40 ± 3	$\mathbf{2.93 \pm 0.01}$	$\mathbf{0.89 \pm 0.01}$ ▼	26 ± 1
	200	2.33 ± 0.04 ◀	0.75 ± 0.01	57 ± 3	$\mathbf{2.31 \pm 0.02}$ ◀	$\mathbf{0.71 \pm 0.01}$	$\mathbf{31 \pm 1}$
	400	1.72 ± 0.03	0.54 ± 0.01 ▶	81 ± 4	$\mathbf{1.53 \pm 0.02}$	$\mathbf{0.52 \pm 0.01}$ ◆ ▶	$\mathbf{51 \pm 1}$
$\mathcal{I}_{\mathrm{FT}}$	0	3.46 ± 0.01	0.95 ± 0.01	15 ± 1	2.78 ± 0.01	0.89 ± 0.01	16 ± 1
	100	$\mathbf{2.81 \pm 0.03}$	$\mathbf{0.90 \pm 0.01}$	$\mathbf{17 \pm 2}$	2.57 ± 0.02	0.81 ± 0.01	$\mathbf{20 \pm 1}$
	200	$\mathbf{2.55 \pm 0.04}$	$\mathbf{0.84 \pm 0.01}$	$\mathbf{24 \pm 3}$	2.23 ± 0.02	0.69 ± 0.01	41 ± 3
	400	$\mathbf{1.97 \pm 0.03}$	$\mathbf{0.79 \pm 0.01}$	$\mathbf{39 \pm 2}$	1.39 ± 0.03	0.51 ± 0.01 ◆	56 ± 3

5.2. Informed inpainting

5.2.1. Objective evaluation

The results of the objective evaluation of informed inpainting in terms of PESQ, STOI and CER scores are presented in Table 1. We assessed here the significance of *mask length* (100 , 200 , 400 ms), *framework* ($\mathcal{I}_{\mathrm{LI}}$, $\mathcal{I}_{\mathrm{ASR}}$, $\mathcal{I}_{\mathrm{PT}}$, $\mathcal{I}_{\mathrm{FT}}$) and *dataset* (single- and multi-speaker) with the test utterances as a random factor for each objective metric. Statistical analysis showed that all factors and all their interactions have a significant effect on each objective metric. Non-significant pairs of distributions shown by post-hoc analyses are indicated by pairs of symbols in Table 1 and reported accordingly in the text.

Influence of mask length. Pairwise comparisons show significant differences between the three *mask length* levels, on all metrics, for each *framework* and each *dataset*, except in terms of STOI between mask lengths of 100 ms and 200 ms in the $\mathcal{I}_{\mathrm{ASR}}$ × single-speaker condition (●); and in terms of CER between mask lengths of 100 ms and 200 ms in the $\mathcal{I}_{\mathrm{PT}}$ × single-speaker condition (■). As expected, as the mask length increases from 100 to 400 ms, the performance across all evaluated metrics decreases. For example, for the $\mathcal{I}_{\mathrm{FT}}$ framework, the PESQ score is 3.28 for a mask length of 100 ms and it drops to 2.93 for a mask length of 400 ms.

Comparison between models and baselines. Pairwise differences between the four inpainting *frameworks* metric distributions are as follows. First, $\mathcal{I}_{\mathrm{PT}}$ and $\mathcal{I}_{\mathrm{FT}}$ display significant differences for all *mask length* and *datasets*, except in terms of STOI in the 400 ms × single-speaker (★) condition; and in terms of CER in the 200 ms × single-speaker (◆) and in the 400 ms × multi-speaker (●) conditions. The nature of these differences depends on the *dataset* and are discussed in the next Section.

Second, the baselines $\mathcal{I}_{\mathrm{LI}}$ and $\mathcal{I}_{\mathrm{ASR}}$ display significant differences for all *mask lengths* and *datasets* in terms of PESQ, except in the 100 ms × single-speaker (■) condition; and for all *mask length* and *datasets* in terms of STOI. By contrast,

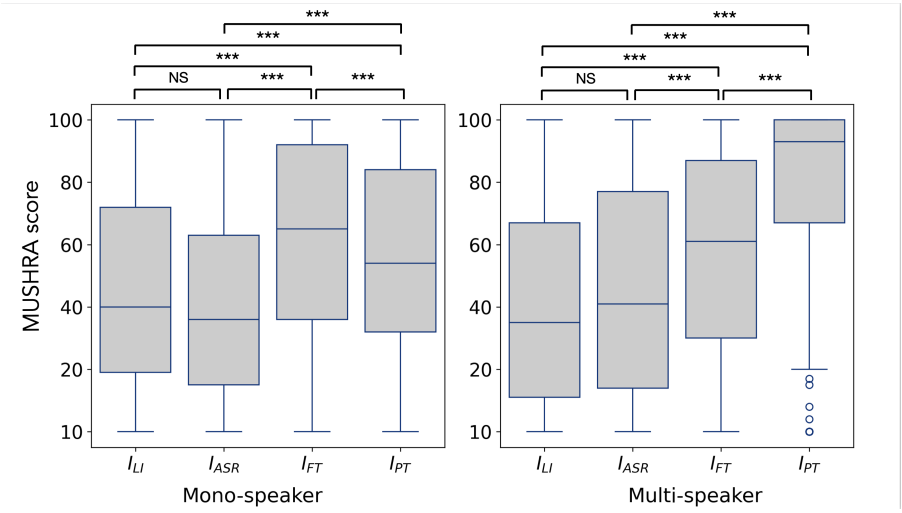


Figure 5: Boxplots illustrating the distribution of the MUSHRA scores for the two models $\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{PT}}$, and for the $\mathcal{I}_{\texttt{LI}}$ and $\mathcal{I}_{\texttt{ASR}}$ baselines (informed inpainting with a 200 ms-length mask). *** indicates that the differences between each pairs of inpainting frameworks were found very significant (i.e. $p \leqslant 0.001$).

they only differ in terms of CER in the 100 ms $\times$ multi-speaker condition. This shows that if $\mathcal{I}_{\texttt{ASR}}$ is superior in terms of quality of the signal reconstruction, both baseline methods are similar in terms of intelligibility of the reconstructed signal.

Third, both proposed inpainting frameworks ($\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{PT}}$) obtain scores that are systematically greater (and often much greater) than those obtained by the $\mathcal{I}_{\texttt{LI}}$ baseline, for both the single-speaker and multi-speaker cases. This confirms the expected need for a non-linear modelling to fill gaps that include more than a diphone transition. This also demonstrates the interest of using a powerful encoder like HuBERT, which is able to exploit the contextual information to access the high-level linguistic information needed for inpainting long gaps.

Finally, both proposed inpainting frameworks ($\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{PT}}$) obtained significantly higher scores than $\mathcal{I}_{\texttt{ASR}}$ in terms of PESQ for all conditions. Both methods also outperform the $\mathcal{I}_{\texttt{ASR}}$ baseline in terms of STOI and CER in the single-speaker setting, expect between the $\mathcal{I}_{\texttt{ASR}}$ and $\mathcal{I}_{\texttt{PT}}$ STOI in the 100 ms $\times$ single-speaker condition (●); between the $\mathcal{I}_{\texttt{ASR}}$ and $\mathcal{I}_{\texttt{FT}}$ STOI in the 200 ms $\times$ single-speaker condition (◆), and between the $\mathcal{I}_{\texttt{ASR}}$ and $\mathcal{I}_{\texttt{PT}}$ CER in the 400 ms $\times$ single-speaker condition (■). In the multi-speaker setting, only $\mathcal{I}_{\texttt{PT}}$ is competitive compared to $\mathcal{I}_{\texttt{ASR}}$, displaying significantly greater score for both STOI and CER in all conditions. These results highlight the effectiveness of the SSL-based approach, particularly $\mathcal{I}_{\texttt{PT}}$. It can be hypothesized that HuBERT successfully infers the linguistic content of missing signal segments from their surrounding context, achieving performance comparable to the ASR used in $\mathcal{I}_{\texttt{ASR}}$, especially in the 100 ms and 200 ms conditions, where the inpainted segments are shorter than a word. However, the performance of both approaches declines for masks of approximately 400 ms, which roughly corresponds to the duration of a word. In such cases, the target word is often omitted in the ASR output, leading to poor signal reconstruction. Additionally, the SSL-based approaches appear to generate fewer audio artifacts compared to $\mathcal{I}_{\texttt{ASR}}$, even when the target text is accurately decoded. This may be due to challenges faced by the zero-shot TTS in accurately replicating the target speaker's timbre, as well as potential alignment errors during the DTW.

Single-speaker vs. multi-speaker. All metrics distributions between *datasets* are also significant for each *framework* and *mask length*, except between the single- and multi-speaker datasets in terms of STOI in the $\mathcal{I}_{\texttt{PT}}$ $\times$ 200 ms (▸) condition; and in terms of CER in the $\mathcal{I}_{\texttt{PT}}$ $\times$ 200 ms (▾) and in the $\mathcal{I}_{\texttt{FT}}$ $\times$ 200 ms (▴) conditions. Interestingly, results display a strong interaction between the *dataset* and the *framework* factors. In the single-speaker case, the $\mathcal{I}_{\texttt{FT}}$ framework (fine-tuned SSL encoder) consistently outperforms the $\mathcal{I}_{\texttt{PT}}$ framework (frozen pre-trained SSL encoder) across all evaluated metrics. For example, for a mask length of 100 ms, $\mathcal{I}_{\texttt{FT}}$ achieves a PESQ score of 3.28, a STOI score of 0.96, and a CER of 7 %, whereas $\mathcal{I}_{\texttt{PT}}$ obtains 3.06, 0.94, and 15 %, respectively. Conversely, in the multi-speaker setting (VCTK dataset), best performances are systematically obtained with $\mathcal{I}_{\texttt{PT}}$. For example, with a mask length of 400 ms, $\mathcal{I}_{\texttt{FT}}$ gets a PESQ score of 2.39, a STOI score of 0.79, and a CER of 19 %, whereas $\mathcal{I}_{\texttt{PT}}$ yields scores of 2.66, 0.83, and 18 % respectively. The difference probably stems from the difficulty for $\mathcal{I}_{\texttt{FT}}$ to compress in a single codebook all the inter-speaker variability. The use of a speaker embedding as done in $\mathcal{I}_{\texttt{PT}}$ appears to be a much more efficient strategy.

5.2.2. Perceptual evaluation

Results of the perceptual evaluation, conducted in the informed case with masks of 200 ms, are presented in Fig. 5. We assessed here the significance of the *framework* ($\mathcal{I}_{\texttt{LI}}$, $\mathcal{I}_{\texttt{ASR}}$, $\mathcal{I}_{\texttt{PT}}$, $\mathcal{I}_{\texttt{FT}}$) factor with the participants as a random effect,

and pairwise comparison displays significant differences between all frameworks. These results confirm all the trends revealed by the objective scores. First, despite its superior performance in terms of sound quality (see PESQ scores), the $\mathcal{I}_{\texttt{ASR}}$ baseline is not significantly preferred by listeners over the $\mathcal{I}_{\texttt{LI}}$ approach. Second, the two SSL-based frameworks $\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{PT}}$ clearly outperform these baselines. The $\mathcal{I}_{\texttt{FT}}$ framework provides better results than the $\mathcal{I}_{\texttt{PT}}$ framework in the single-speaker case, and the opposite is observed in the multi-speaker case (with an even more marked difference between the two frameworks). Notably, the $\mathcal{I}_{\texttt{PT}}$ framework achieves a high performance level in the multi-speaker case, exceeding 90%, with the reconstructed signal closely resembling the original.

5.3. Blind inpainting

The results of the objective evaluation of blind inpainting in terms of PESQ, STOI and CER scores are presented in Table 2. To compare informed vs. blind inpainting, we assessed the significance of *mask length* (100, 200, 400 ms), *framework* ($\mathcal{I}_{\texttt{PT}}$, $\mathcal{I}_{\texttt{FT}}$), *dataset* (single- and multi-speaker) and *type of inpainting* (informed vs. blind) with the test utterances as a random factor for each objective metric. Note that compared to Section 5.2.1, the $\mathcal{I}_{\texttt{LI}}$ and $\mathcal{I}_{\texttt{ASR}}$ levels are removed from the *framework* factor, as they are not evaluated in the blind inpainting case. Statistical analysis shows that all factors and all their interactions have a significant effect on each objective metric.

Informed vs. blind. Pairwise comparisons show significant differences between the informed and blind metrics distributions, for each *framework*, *dataset*, and *mask length*. Compared to the informed inpainting configuration, the blind configuration is more challenging (the position of the mask is unknown and the full signal hence is reconstructed). As expected, it leads to lower performance, and this is observed for both $\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{PT}}$, for both datasets, all mask lengths, and all metrics. For example, for blind inpainting with a 200 ms mask length in the single-speaker case, $\mathcal{I}_{\texttt{FT}}$ gets a STOI score of 0.84, compared to 0.93 in the corresponding informed case. Moreover, informed inpainting methods consistently exhibit lower CER, reflecting higher accuracy in reconstructing corrupted segments.

Effect of mask length, framework, and dataset. All pairs of distributions across the three factors are significant, except in terms of PESQ between single- and multi-speaker datasets in the $\mathcal{I}_{\texttt{PT}}$ $\times$ 200 ms condition (◂); in terms of STOI between the $\mathcal{I}_{\texttt{PT}}$ and $\mathcal{I}_{\texttt{FT}}$ frameworks in the 400 ms $\times$ multi-speaker condition (◆) and between single- and multi-speaker in the $\mathcal{I}_{\texttt{PT}}$ $\times$ 100 ms (▾) and $\mathcal{I}_{\texttt{PT}}$ $\times$ 400 ms (▸) conditions. Interestingly, the interactions between the *type of inpainting* and each of these three factors are weak, as all the trends observed in the informed inpainting case remain in the blind inpainting case. Similarly to informed inpainting, performances of all metrics drop as mask length increases. Above all, the interaction between *framework* and *dataset* is still present, as the $\mathcal{I}_{\texttt{FT}}$ framework provides better results than the $\mathcal{I}_{\texttt{PT}}$ framework in the single-speaker case, and the opposite is observed in the multi-speaker case.

5.4. Comparison with other studies

As announced in Section 4.3, we compare the overall performance of the proposed frameworks with that of recently published methods (Kegler et al., 2020; Zhao, 2023; Xue et al., 2023; Zhang et al., 2024a). We recall that, since no source code was available for these techniques, we use the scores reported in the papers, and we compare the performances only in terms of order of magnitudes.

Comparison with supervised methods. We start with (Kegler et al., 2020) and (Zhao, 2023), which are both based on deep supervised learning, as they break down PESQ and STOI performance per *mask length* and thus allow a direct comparison with our results. In (Kegler et al., 2020), with a training and test on a multi-speaker dataset (LibriSpeech) and in the informed inpainting case, the authors reported a PESQ (resp. STOI) score of 3.24, 2.81, and 2.18 (resp. 0.94, 0.89, and 0.73) for mask lengths of 100, 200, and 400 ms, respectively. In (Zhao, 2023), the authors reported PESQ (resp. STOI) scores of 3.30, 2.61, and 1.76 (resp. 0.96, 0.89, and 0.73) for similar masks and dataset. In our study, the best scores on the same settings for $\mathcal{I}_{\texttt{PT}}$ (resp. $\mathcal{I}_{\texttt{FT}}$), were 3.13, 2.93, and 2.66 (resp. 3.06, 2.70, and 2.39) for PESQ, and 0.93, 0.88, and 0.83 (resp. 0.90, 0.85, and 0.79) for STOI (see Table 1, multi-speaker).

For the blind case, we can only compare our results to those reported in (Kegler et al., 2020) (Table 3, condition "FC-gaps") since it is not treated in (Zhao, 2023), to the best of our understanding. In this case, our performances are significantly lower, both in terms of STOI and PESQ. For example, for a mask length of 400 ms (Kegler et al., 2020) reported a quite high STOI score of 0.71 when we obtained only 0.52 with the (best) framework $\mathcal{I}_{\texttt{PT}}$. The differences between the two techniques are smaller when the mask length is shorter (e.g. a PESQ score of 2.72 in (Kegler et al., 2020) for a mask length of 200 ms vs. 2.31 with $\mathcal{I}_{\texttt{PT}}$). Further experiments could be useful to better understand the origin of these differences in the case of blind inpainting. We would need to check that this is not simply due to the nature of the training/test datasets, to the analysis-synthesis ability of the methods, or to a different way of calculating the PESQ and STOI scores.

Comparison with self-supervised methods. We now have a look at (Xue et al., 2023) and (Zhang et al., 2024a), as both methods used an SSL encoder, but fine-tuned to the task of inpainting. Comparison is more difficult with those studies, as experimental conditions are different from ours: they evaluate PLC, with several gaps per signal, while we evaluate mainstream inpainting with a single gap. Zhang et al. (2024a) reported a PESQ (resp. STOI) score of 3.27 and 2.35 (resp. 0.93 and 0.80) for "easy" (15 % loss) and "medium" (30 % loss) conditions on a multi-speaker corpus (VCTK), in a blind setting. Our performance with $\mathcal{I}_{\texttt{PT}}$ is lower with both 10 % loss (PESQ of 2.93 and STOI of 0.89) compared to their easy condition, and 40 % loss (PESQ of 1.53 and STOI of 0.52) compared to their "medium" condition. Xue et al. (2023) reached an overall PESQ score of 2.88 with losses ranging from 10 % to 50 % in a blind setting, which is in the order of magnitude of our PESQ score with 10 % loss (PESQ of 2.93). Before drawing any conclusions, it should be reminded that the loss percentage reported by these studies accumulates the length of smaller gaps introduced into speech signals, and that the total length of those signals—accounting for the amount of uncorrupted speech—is not given by the authors. In our study, the loss corresponds to a single gap within a one-second signal. As a result, it is not clear which configuration is the most challenging, i.e., if the amount and position of uncorrupted portions of the input signal on which the SSL encoder relies is comparable between methods, and thus if those results could be compared with ours on an equal footing.

To conclude, this "meta-comparison" shows that the two proposed frameworks $\mathcal{I}_{\texttt{FT}}$ and $\mathcal{I}_{\texttt{PT}}$ seem to outperform other approaches based on supervised learning, at least in the informed case, and in particular for long masks (i.e., 400 ms). Here again, this can be explained by the ability of a powerful SSL model, pre-trained on a large amount of data, to extract the high-level linguistic information (e.g., syntactic and semantic) of the sentence to be reconstructed, based on the contextual non-missing information. Nevertheless, although the results should be treated with caution, it seems that when SSL encoders are introduced, the introduction of auxiliary modules and losses dedicated to the inpainting task are beneficial to model performance.

5.5. Extrapolation to other datasets capabilities

We now explore the extrapolation capabilities of our pre-trained framework $\mathcal{I}_{\texttt{PT}}$ to out-of-domain datasets, in the informed case. We omitted here the $\mathcal{I}_{\texttt{LI}}$ and $\mathcal{I}_{\texttt{ASR}}$ baselines which presented the lowest scores in previous evaluations.

Evaluation on multi-speaker expressive speech. Table 3a reports the results of the objective evaluation in terms of PESQ, STOI and CER scores. We assessed the significance of *mask length* (100 , 200 , 400 ms) with *utterance* as a random factor. Statistical analysis shows that *mask length* had a significant effect on each objective metric, and the *utterance* random factor had no significant effect on PESQ and was therefore removed from this model. Pairwise comparisons show significant differences between the three *mask length* levels, on all metrics, corresponding to a decrease in performance with increasing mask lengths.

Although we observe a drop in all scores when tested on the expressive speech data compared to the evaluation on the in-domain VCTK test set (Table 1), $\mathcal{I}_{\texttt{PT}}$ still achieves higher scores than work reported in other studies (Kegler et al., 2020; Zhao, 2023) which only performed in-domain testing (see Section 5.4). Therefore, this demonstrates the robustness of our framework to zero-shot adaptation to both unseen speakers and unseen speaking style in the particular case of expressive speech synthesis.

Evaluation on noise-corrupted speech. Table 3b reports the results of the objective evaluation of $\mathcal{I}_{\texttt{PT}}$ in terms of PESQ and STOI scores for white noise and crowd noise corruption, respectively. We assessed here the significance of *mask length* (100 , 200 , 400 ms), *SNR* (10 , 20 , 40 dB), *noise type* (white noise and crowd noise) and *dataset* (single- and multi-speaker) with their interactions.

Statistical analysis of PESQ distributions showed that the four-way interaction (*mask length* × *SNR* × *noise* × *dataset*) as well as all three-way and two-way interactions were not significant. This allowed to simplify the statistical model by removing one interaction at a time while assessing if the removal of each interaction had a significant impact on the model. Ultimately, only the *mask length* and *SNR* were found to have a significant effect on the PESQ distributions. The effect *mask length* shows the same trends as in clean speech, with a decrease of performance with longer masks. Then, the global decrease of PESQ across all conditions with decreasing *SNR* displays a sensitivity to the amount of corruption noise that is introduced. Interestingly, the non-significance of the *noise type* factor indicates that the $\mathcal{I}_{\texttt{PT}}$ method is equally robust to either type of noise. Similarly, the absence of a *dataset* effect indicates that the addition of noise has levelled scores to a point that mitigate the impact of the training corpus on the model.

Statistical analysis of STOI distributions also gave a simple model, where only the *mask length* and the *SNR* × *dataset* interaction have a significant effect. We find again a decreasing of STOI with longer masks. As for the interaction, multiple comparisons show that the effect of *SNR* is only present in the multi-speaker condition, while intelligibility is little affected by the amount of corruption noise in the single-speaker condition. Compared to PESQ, we found a significant effect of *datasets* for SNRs of 10 and 20 dB only. Therefore, the differences in intelligibility scores observed on clean speech between the single-speaker and the multi-speaker *datasets* remain up to the highest level of corruption noise. The absence of effect of the *noise type* factor again shows that all observations hold for both white noise and crowd noise.

Table 3: Informed speech inpainting results with the $\mathcal{I}_{\text{PT}}$ framework on the out-of-domain test sets. For all metrics, average scores with confidence intervals for each test set and each mask length for both the single- and multi-speaker configurations.

(a) Results on expressive speech (Expresso dataset).

		Multi-speaker (Expresso)		
Models	Mask (ms)	PESQ $[-0.5; 4.5] \uparrow$	STOI $[0; 1] \uparrow$	CER (%) $\downarrow$
Unmasked	0	4.07 ± 0.01	0.96 ± 0.01	8 ± 1
$\mathcal{I}_{\text{PT}}$	100	3.02 ± 0.03	0.91 ± 0.01	13 ± 1
	200	2.89 ± 0.03	0.84 ± 0.01	18 ± 1
	400	2.64 ± 0.03	0.80 ± 0.01	24 ± 2

(b) Results on the noise-corrupted test sets. Results are broken down for each SNR and noise type.

			Single-speaker (LJ Speech)		**Multi-speaker (VCTK)**	
Noise type	Mask (ms)	SNR (dB)	PESQ $[-0.5; 4.5] \uparrow$	STOI $[0; 1] \uparrow$	PESQ $[-0.5; 4.5] \uparrow$	STOI $[0; 1] \uparrow$
White noise	100	20	3.20 ± 0.32	0.91 ± 0.02	3.47 ± 0.26	0.96 ± 0.02
		10	2.79 ± 0.23	0.89 ± 0.02	3.18 ± 0.25	0.95 ± 0.03
		0	2.95 ± 0.38	0.85 ± 0.05	2.98 ± 0.30	0.91 ± 0.02
	200	20	2.84 ± 0.28	0.87 ± 0.03	3.04 ± 0.36	0.94 ± 0.03
		10	2.74 ± 0.34	0.81 ± 0.05	3.03 ± 0.38	0.88 ± 0.04
		0	2.72 ± 0.25	0.85 ± 0.06	2.80 ± 0.34	0.87 ± 0.05
	400	20	2.75 ± 0.26	0.82 ± 0.04	2.88 ± 0.34	0.91 ± 0.05
		10	2.43 ± 0.27	0.78 ± 0.06	2.65 ± 0.33	0.87 ± 0.06
		0	2.27 ± 0.30	0.78 ± 0.05	2.30 ± 0.36	0.79 ± 0.06
Crowd noise	100	20	3.13 ± 0.27	0.91 ± 0.02	3.35 ± 0.29	0.96 ± 0.02
		10	3.01 ± 0.28	0.89 ± 0.03	3.15 ± 0.29	0.94 ± 0.02
		0	2.82 ± 0.31	0.86 ± 0.04	2.94 ± 0.26	0.91 ± 0.03
	200	20	2.96 ± 0.30	0.86 ± 0.03	3.12 ± 0.33	0.94 ± 0.03
		10	2.74 ± 0.31	0.85 ± 0.04	2.91 ± 0.32	0.91 ± 0.03
		0	2.58 ± 0.29	0.84 ± 0.06	2.71 ± 0.32	0.84 ± 0.04
	400	20	2.81 ± 0.27	0.84 ± 0.04	2.88 ± 0.35	0.90 ± 0.04
		10	2.50 ± 0.33	0.82 ± 0.05	2.75 ± 0.32	0.87 ± 0.05
		0	2.31 ± 0.32	0.79 ± 0.05	2.51 ± 0.34	0.80 ± 0.06

To conclude, we observed very similar PESQ and STOI scores between white noise- and crowd noise-corrupted test sets, supported by the non significance of the *noise type* factor. Moreover, it seems that the addition of corruption noise has affected both PESQ and STOI performances to the point where most other factors have less impact on the objective measures. In the end, $\mathcal{I}_{\text{PT}}$ PESQ scores are even comparable with those obtained on clean speech for both corruption noises (Table 1). This proves a high robustness of this model to corrupted input signal. Last but not least, it is important to note that the inpainting of noise-corrupted signals generates a portion of signal which is without noise. Thus, while this experiment is valuable to demonstrate the robustness of our inpainting method to the corruption of the signal used as context, the resulting output signal is not currently relevant for a direct application to a speech generation use case.

6. Conclusion

This study investigated the extent to which contextual speech representations extracted by an SSL model can effectively drive an inpainting process, i.e., reconstructing a missing portion of a speech signal from its surrounding context. More specifically, our hypothesis was that it is possible to perform speech inpainting with an SSL encoder that is pre-trained on an unmasking task, only by appending a decoder to convert back to the time domain. We compared two different frameworks for this sake. First, a pre-trained HuBERT SSL encoder with an adapted HiFi-GAN decoder ($\mathcal{I}_{\text{PT}}$) was used to test our hypothesis. Second, a fine-tuned HuBERT SSL encoder to the inpainting task with a pre-trained vanilla HiFi-GAN ($\mathcal{I}_{\text{FT}}$) was used as a reference to state-of-the-art methods that all train their model specifically for the inpainting task. Additionally, we explored an alternative approach inspired by text-informed inpainting, which combines

ASR with zero-shot TTS ($\mathcal{I}_{\texttt{ASR}}$). We evaluated our main method on both in-domain and out-of-domain datasets to evaluate its robustness to a variety of input signals.

As a main result, we validated our main hypothesis, by demonstrating that a pre-trained SSL-based approach ($\mathcal{I}_{\texttt{PT}}$) can successfully perform inpainting tasks both in terms of quality and intelligibility, as measured with both objective and perceptive metrics. $\mathcal{I}_{\texttt{PT}}$ systematically outperformed our baselines, including $\mathcal{I}_{\texttt{ASR}}$ which performed inpainting using a text-based language model. Interestingly, when compared to other studies in the literature, $\mathcal{I}_{\texttt{PT}}$ was extremely competitive with the most recent supervised methods, while our method was not specifically trained for the inpainting task, but indirectly trained on unmasking. This clearly demonstrates the potential of SSL encoders to transfer knowledge between unmasking and inpainting without the need for fine-tuning, provided that a decoder is appended to generate a speech waveform. Moreover, we showed that such method is resilient to varied input data including expressive speech and noise-corrupted speech—two challenging conditions that, to the best of our knowledge, have received limited attention in previous inpainting studies.

Then, we evaluated the benefit of fine-tuning the SSL encoder to the inpainting task. We observed a symmetry in our results, with fine-tuning the SSL encoder ($\mathcal{I}_{\texttt{FT}}$) being the most effective strategy for inpainting in a single-speaker setting, whereas the pre-trained encoder ($\mathcal{I}_{\texttt{PT}}$) yielded better results in a multi-speaker scenario. The higher performance of $\mathcal{I}_{\texttt{FT}}$ on the single-speaker supports the hypothesis that while a pre-trained encoder is already successful in inpainting, the addition of further supervised training is beneficial to the task. In the multi-speaker setting, one main difference between models is the absence of a speaker embedding when fine-tuning the SSL encoder ($\mathcal{I}_{\texttt{FT}}$). This suggests that the HuBERT SSL encoder may not keep enough speaker information, even in a supervised inpainting learning task, to reconstruct the signal with similar quality than in the pre-trained configuration. Alternate solutions for fine-tuning that put the SSL encoder in a more complex architecture comprising auxiliary modules and varied training losses, such as (Xue et al., 2023; Zhang et al., 2024a), seem to be relevant when looking at performance scores reported in those paper. However, their experimental set-ups are not fully comparable with ours.

Finally, while specifically fine-tuning models on inpainting tasks in a supervised fashion is and will certainly remain the most efficient way to increase performance, we took a step aside in this paper. We followed a more hypothesis-driven approach based on a controlled and symmetric experimental protocol (pre-trained vs. fine-tuned SSL) to quantitatively assess the relationships between the pretext task of SSL (unmasking) and the inpainting downstream task. By demonstrating successful transfer between the two tasks, this paves the way for further use of this simple single SSL encoder configuration for inpainting, as a tool for understanding the speech inpainting process and its relationship to language and cognition. In particular, future work may focus on (i) benchmarking other SSL models for inpainting, including models not fine-tuned for ASR, as well as alternatives such as wav2vec (Baevski et al., 2020), and WavLM (Chen et al., 2022); (ii) a fine-grained analysis of the inpainted speech at different linguistic scales (phonetic, syllabic, morphologic); (iii) performing an in-depth study of the impact of the codebook size C on overall performance; and (iv) the relationship between the context actually used by the SSL encoder on one hand, and the length and linguistic complexity of the signal to be reconstructed, on the other. Finally, in addition to their technological applications, the proposed speech inpainting systems, and SSL models in general, provide a means of finely quantifying the amount of predictable information in the speech signal. Therefore, they can be potentially useful for studying, through computational modelling and simulation, some of the predictive processes underlying speech perception (Friston and Kiebel, 2009; Tavano and Scharinger, 2015). The proposed framework based on non-causal prediction could complement other studies conducted in the context of the predictive coding framework and focusing on causal predictions (e.g., (Hueber et al., 2020; Caucheteux et al., 2023; Heilbron et al., 2019)).

Acknowledgment

This work has been partially supported by MIAI Cluster (ANR-23-IACL-0006), MIAI@Grenoble Alpes, (ANR-19-P3IA-0003), and by ANRT CIFRE.

CRediT authorship contribution statement

Ihab Asaad: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft. **Maxime Jacquelin:** Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft. **Olivier Perrotin:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing, Visualization, Supervision, Funding acquisition. **Laurent Girin:** Conceptualization, Formal analysis, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition. **Thomas Hueber:** Conceptualization, Methodology, Software, Validation, Investigation, Resources, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition, Project administration.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used *DeepL* and *Le Chat - Mistral AI* in order to polish the English of some paragraphs. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

Aironi, C., Gabrielli, L., Cornell, S., Squartini, S., 2025. Complex-bin2bin: A latency-flexible generative neural model for audio packet loss concealment. IEEE Transactions on Audio, Speech and Language Processing 33, 199–210. doi:10.1109/TASLP.2024.3515794.

Arora, S., Chang, K.W., Chien, C.M., Peng, Y., Wu, H., Adi, Y., Dupoux, E., Lee, H.Y., Livescu, K., Watanabe, S., 2025. On the landscape of spoken language models: A comprehensive survey. arXiv preprint arXiv:2504.08528 .

Baevski, A., Zhou, Y., Mohamed, A., Auli, M., 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations, in: Advances in Neural Information Processing Systems, Vancouver, Canada. pp. 12449–12460. URL: https://dl.acm.org/doi/abs/10.5555/3495724.3496768.

Borgström, B.J., Borgström, P.H., Alwan, A., 2010. Efficient HMM-based estimation of missing features, with applications to packet loss concealment., in: Interspeech, Makuhari, Chiba, Japan. pp. 2394–2397. doi:10.21437/Interspeech.2010-654.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., Zeghidour, N., 2023. AudioLM: a language modeling approach to audio generation. IEEE/ACM Transactions on Audio, Speech, and Language Processing 31, 2523–2533. doi:10.1109/TASLP.2023.3288409.

Borsos, Z., Sharifi, M., Tagliasacchi, M., 2022. Speechpainter: Text-conditioned speech inpainting, in: Interspeech, Incheon, Korea. pp. 431–435. doi:10.21437/Interspeech.2022-194.

Casanova, E., Weber, J., Shulby, C.D., Junior, A.C., Gölge, E., Ponti, M.A., 2022. YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone, in: Proc. of the International Conference on Machine Learning, PMLR, Baltimore, Maryland, USA. pp. 2709–2720. URL: https://proceedings.mlr.press/v162/casanova22a.html.

Caucheteux, C., Gramfort, A., King, J.R., 2023. Evidence of a predictive coding hierarchy in the human brain listening to speech. Nature human behaviour 7, 430–441. doi:10.1038/s41562-022-01516-2.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., Yu, X., Wei, F., 2022. WavLM: Large-scale self-supervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing 16, 1505–1518. doi:10.1109/JSTSP.2022.3188113.

Chen, S., Wang, C., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., He, L., Zhao, S., Wei, F., 2025. Neural codec language models are zero-shot text to speech synthesizers. IEEE Transactions on Audio, Speech and Language Processing 33, 705–718. doi:10.1109/TASLPRO.2025.3530270.

Dai, L., Ke, Y., Zhang, H., Hao, F., Luo, X., Li, X., Zheng, C., 2024. A time-frequency band-split neural network for real-time full-band packet loss concealment, in: IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), pp. 77–78. doi:10.1109/ICASSPW62465.2024.10626621.

Diener, L., Branets, S., Saabas, A., Cutler, R., 2025. The ICASSP 2024 audio deep packet loss concealment grand challenge. IEEE Open Journal of Signal Processing 6, 231–237. doi:10.1109/OJSP.2025.3526552.

Diener, L., Sootla, S., Branets, S., Saabas, A., Aichner, R., Cutler, R., 2022. INTERSPEECH 2022 audio deep packet loss concealment challenge, in: Interspeech, Incheon, South Korea. pp. 580–584. doi:10.21437/Interspeech.2022-10829.

Du, C., Guo, Y., Shen, F., Liu, Z., Liang, Z., Chen, X., Wang, S., Zhang, H., Yu, K., 2024. UniCATS: a unified context-aware text-to-speech framework with contextual VQ-diffusion and vocoding. Proceedings of the AAAI Conference on Artificial Intelligence 38, 17924–17932. doi:10.1609/aaai.v38i16.29747.

Du, Z., Zhang, X., Han, J., 2020. A joint framework of denoising autoencoder and generative vocoder for monaural speech enhancement. IEEE/ACM Transactions on Audio, Speech, and Language Processing 28, 1493–1505. doi:10.1109/TASLP.2020.2991537.

Etter, W., 1996. Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters. IEEE Transactions on Signal Processing 44, 1124–1135. doi:10.1109/78.502326.

Friston, K., Kiebel, S., 2009. Predictive coding under the free-energy principle. Philosophical transactions of the Royal Society B: Biological sciences 364, 1211–1221. doi:10.1098/rstb.2008.0300.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in Neural Information Processing Systems, Montreal, Canada. pp. 2672–2680. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf.

Gunduzhan, E., Momtahan, K., 2001. Linear prediction based packet loss concealment algorithm for pcm coded speech. IEEE Transactions on Speech and Audio Processing 9, 778–785. doi:10.1109/89.966081.

Heigold, G., Moreno, I., Bengio, S., Shazeer, N., 2016. End-to-end text-dependent speaker verification, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China. pp. 5115–5119. doi:10.1109/ICASSP.2016.7472652.

Heilbron, M., Ehinger, B.V., Hagoort, P., de Lange, F.P., 2019. Tracking naturalistic linguistic predictions with deep neural language models, in: Proc. of the Conference on Cognitive Computational Neuroscience, Berlin, Germany. pp. 424–427. doi:10.32470/CCN.2019.1096-0.

Hsu, W.N., Bolte, B., Tsai, Y.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A., 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Transactions on Audio, Speech, and Language Processing 29, 3451–3460. doi:10.1109/TASLP.2021.3122291.

Hsu, W.N., Remez, T., Shi, B., Donley, J., Adi, Y., 2023. ReVISE: self-supervised speech resynthesis with visual input for universal and generalized speech regeneration, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada. pp. 18796–18806. doi:10.1109/CVPR52729.2023.01802.

Hueber, T., Tatulli, E., Girin, L., Schwartz, J.L., 2020. Evaluating the Potential Gain of Auditory and Audiovisual Speech-Predictive Coding Using Deep Learning. Neural Computation 32, 596–625. doi:10.1162/neco_a_01264.

Irvin, B., Stamenovic, M., Kegler, M., Yang, L.C., 2023. Self-supervised learning for speech enhancement through synthesis, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece. pp. 1–5. doi:10.1109/ICASSP49357.2023.10094705.

Ito, K., Johnson, L., 2017. The LJ Speech Dataset. https://keithito.com/LJ-Speech-Dataset/.

ITU, 2015. Method for the subjective assessment of intermediate quality level of audio systems. Technical Report ITU-R BS.1534-3. International Telecommunication Union. URL: https://www.itu.int/rec/R-REC-BS.1534.

Jillings, N., Moffat, D., De Man, B., Reiss, J.D., 2015. Web Audio Evaluation Tool: A browser-based listening test environment, in: Proc. of the Sound and Music Computing Conference, Maynooth, Ireland. URL: https://code.soundsoftware.ac.uk/projects/webaudioevaluationtool.

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., Likhomanenko, T., Synnaeve, G., Joulin, A., Mohamed, A., Dupoux, E., 2020. Libri-Light: A benchmark for ASR with limited or no supervision, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain. pp. 7669–7673. doi:10.1109/ICASSP40776.2020.9052942. https://github.com/facebookresearch/libri-light.

Kegler, M., Beckmann, P., Cernak, M., 2020. Deep speech inpainting of time-frequency masks, in: Interspeech, Shanghai, China. pp. 3276–3280. doi:10.21437/Interspeech.2020-1532.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: International Conference on Learning Representations (ICLR), San Diego, CA, USA.

Koizumi, Y., Zen, H., Karita, S., Ding, Y., Yatabe, K., Morioka, N., Zhang, Y., Han, W., Bapna, A., Bacchiani, M., 2023. Miipher: A robust speech restoration model integrating self-supervised speech and text representations, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA. pp. 1–5. doi:10.1109/WASPAA58266.2023.10248089.

Kondo, K., Nakagawa, K., 2006. A speech packet loss concealment method using linear prediction. IEICE - Trans. Inf. Syst. E89-D, 806—813. doi:10.1093/ietisy/e89-d.2.806.

Kong, J., Kim, J., Bae, J., 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. Advances in Neural Information Processing Systems 33, 17022–17033. URL: https://dl.acm.org/doi/abs/10.5555/3495724.3497152.

Lagrange, M., Marchand, S., Rault, J.B., 2005. Long interpolation of audio signals using linear prediction in sinusoidal modeling. Journal of the Audio Engineering Society 53, 891–905. URL: https://secure.aes.org/forum/pubs/journal/?elib=13390.

Lakhotia, K., Kharitonov, E., Hsu, W.N., Adi, Y., Polyak, A., Bolte, B., Nguyen, T.A., Copet, J., Baevski, A., Mohamed, A., et al., 2021. On generative spoken language modeling from raw audio. Transactions of the Association for Computational Linguistics 9, 1336–1354. doi:10.1162/tacl_a_00430.

Lee, B.K., Chang, J.H., 2016. Packet loss concealment based on deep neural networks for digital speech transmission. IEEE/ACM Transactions on Audio, Speech, and Language Processing 24, 378–387. doi:10.1109/TASLP.2015.2509780.

Li, N., Zheng, X., Zhang, C., Guo, L., Yu, B., 2022. End-to-end multi-loss training for low delay packet loss concealment, in: Interspeech, Incheon, South Korea. pp. 585–589. doi:10.21437/Interspeech.2022-11439.

Li, W., Bao, C., Zhou, J., Yang, X., 2023. A time-domain packet loss concealment method by designing transformer-based convolutional recurrent network, in: IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Zhengzhou, China. pp. 1–5. doi:10.1109/ICSPCC59353.2023.10400230.

Lin, J., Wang, Y., Kalgaonkar, K., Keren, G., Zhang, D., Fuegen, C., 2021. A time-domain convolutional recurrent network for packet loss concealment, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada. pp. 7148–7152. doi:10.1109/ICASSP39728.2021.9413595.

Lindblom, J., Hedelin, P., 2002. Packet loss concealment based on sinusoidal extrapolation, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, FL, USA. pp. I–173–I–176. doi:10.1109/ICASSP.2002.5743682.

Liu, B., Song, Q., Yang, M., Yuan, W., Wang, T., 2022. PLCNet: real-time packet loss concealment with semi-supervised generative adversarial network, in: Interspeech, Incheon, South Korea. pp. 575–579. doi:10.21437/Interspeech.2022-10428.

Liu, H., Kong, Q., Tian, Q., Zhao, Y., Wang, D., Huang, C., Wang, Y., 2021. VoiceFixer: Toward general speech restoration with neural vocoder. arXiv preprint arXiv:2109.13731 URL: https://openreview.net/forum?id=G-7GlfTneYg.

Loizou, P.C., 2007. Speech enhancement: theory and practice. CRC press. doi:10.1201/9781420015836.

Lotfidereshgi, R., Gournay, P., 2018. Speech prediction using an adaptive recurrent neural network with application to packet loss concealment, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada. pp. 5394–5398. doi:10.1109/ICASSP.2018.8462185.

Maiti, S., Mandel, M.I., 2019. Parametric resynthesis with neural vocoders, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Platz, NY, USA. pp. 303–307. doi:10.1109/WASPAA.2019.8937165.

Mohamed, A., Lee, H.y., Borgholt, L., Havtorn, J.D., Edin, J., Igel, C., Kirchhoff, K., Li, S.W., Livescu, K., Maaløe, L., Sainath, T.N., Watanabe, S., 2022. Self-supervised speech representation learning: A review. IEEE Journal of Selected Topics in Signal Processing 16, 1179–1210. doi:10.1109/JSTSP.2022.3207050.

Mohamed, M.M., Schuller, B.W., 2020. ConcealNet: An end-to-end neural network for packet loss concealment in deep speech emotion recognition. arXiv preprint arXiv:2005.07777 .

Morrone, G., Michelsanti, D., Tan, Z.H., Jensen, J., 2021. Audio-visual speech inpainting with deep learning, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada. pp. 6653–6657. doi:10.1109/ICASSP39728.2021.9413488.

Nguyen, T.A., Hsu, W.N., d'Avirro, A., Shi, B., Gat, I., Fazel-Zarani, M., Remez, T., Copet, J., Synnaeve, G., Hassid, M., et al., 2023a. Expresso: A benchmark and analysis of discrete expressive speech resynthesis, in: Interspeech, Dublin, Ireland. pp. 4823–4827. doi:10.21437/Interspeech.2023-1905.

Nguyen, V.A., Nguyen, A.H.T., Khong, A.W.H., 2023b. Improving performance of real-time full-band blind packet-loss concealment with predictive network, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece. pp. 1–5. doi:10.1109/ICASSP49357.2023.10097132.

Pasad, A., Shi, B., Livescu, K., 2023. Comparative layer-wise analysis of self-supervised speech models, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece. pp. 1–5. doi:10.1109/ICASSP49357.2023.10096149.

Pascual, S., Serrà, J., Pons, J., 2021. Adversarial auto-encoding for packet loss concealment, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA. pp. 71–75. doi:10.1109/WASPAA52581.2021.9632730.

Perkins, C., Hodson, O., Hardman, V., 1998. A survey of packet loss recovery techniques for streaming audio. IEEE Network 12, 40–48. doi:10.1109/65.730750.

Perraudin, N., Holighaus, N., Majdak, P., Balazs, P., 2018. Inpainting of long audio segments with similarity graphs. IEEE/ACM Transactions on Audio, Speech, and Language Processing 26, 1079–1090. doi:10.1109/TASLP.2018.2809864.

Perrotin, O., Stephenson, B., Gerber, S., Bailly, G., King, S., 2025. Refining the evaluation of speech synthesis: A summary of the blizzard challenge 2023. Computer Speech & Language 90, 101747. doi:10.1016/j.csl.2024.101747.

Polyak, A., Adi, Y., Copet, J., Kharitonov, E., Lakhotia, K., Hsu, W.N., Mohamed, A., Dupoux, E., 2021. Speech Resynthesis from Discrete Disentangled Self-Supervised Representations, in: Interspeech, Brno, Czechia. pp. 3615–3619. doi:10.21437/Interspeech.2021-475.

Prablanc, P., Ozerov, A., Duong, N.Q.K., Pérez, P., 2016. Text-informed speech inpainting via voice conversion, in: Proc. of European Signal Processing Conference (EUSIPCO), Budapest, Hungary. pp. 878–882. doi:10.1109/EUSIPCO.2016.7760374.

Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I., 2023. Robust speech recognition via large-scale weak supervision, in: Proc. of the International Conference on Machine Learning (ICML), PMLR, Honolulu, Hawai, USA. pp. 28492–28518. doi:https://dl.acm.

org/doi/10.5555/3618408.3619590.

Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P., 2001. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Salt Lake City, UT, USA. pp. 749–752. doi:10.1109/ICASSP.2001.941023.

Rodbro, C.A., Murthi, M.N., Andersen, S.V., Jensen, S.H., 2006. Hidden markov model-based packet loss concealment for voice over ip. IEEE Transactions on Audio, Speech, and Language Processing 14, 1609–1623. doi:10.1109/TSA.2005.858561.

Saeki, T., Takamichi, S., Nakamura, T., Tanji, N., Saruwatari, H., 2022. SelfRemaster: self-supervised speech restoration with analysis-by-synthesis approach using channel modeling, in: Interspeech, Incheon, Korea. pp. 4406–4410. doi:10.21437/Interspeech.2022-298.

Schneider, S., Baevski, A., Collobert, R., Auli, M., 2019. wav2vec: Unsupervised Pre-Training for Speech Recognition, in: Interspeech, Graz, Austria. pp. 3465–3469. doi:10.21437/Interspeech.2019-1873.

Shi, Y., Zheng, N., Kang, Y., Rong, W., 2019. Speech loss compensation by generative adversarial networks, in: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China. pp. 347–351. doi:10.1109/APSIPAASC47483.2019.9023132.

Su, J., Jin, Z., Finkelstein, A., 2021. Hifi-gan-2: Studio-quality speech enhancement via generative adversarial networks conditioned on acoustic features, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA. pp. 166–170. doi:10.1109/WASPAA52581.2021.9632770.

Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J., 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Dallas, TX, USA. pp. 4214–4217. doi:10.1109/ICASSP.2010.5495701.

Tavano, A., Scharinger, M., 2015. Prediction in speech and language processing. Cortex 68, 1–7. doi:10.1016/j.cortex.2015.05.001.

Thirunavukkarasu, E.S., Karthikeyan, E., 2015. A survey on VoIP packet loss techniques. Int. J. Commun. Netw. Distrib. Syst. 14, 106—-116. doi:10.1504/IJCNDS.2015.066029.

Valin, J.M., Mustafa, A., Montgomery, C., Terriberry, T.B., Klingbeil, M., Smaragdis, P., Krishnaswamy, A., 2022. Real-time packet loss concealment with mixed generative and predictive model, in: Interspeech, Incheon, South Korea. pp. 570–574. doi:10.21437/Interspeech.2022-903.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I., 2017. Attention is all you need, in: Advances in Neural Information Processing Systems. Long Beach, California, USA. volume 30, pp. 5998–6008. URL: https://dl.acm.org/doi/10.5555/3295222.3295349.

Verhelst, W., Roelands, M., 1993. An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Minneapolis, MN, USA. pp. 554–557. doi:10.1109/ICASSP.1993.319366.

Wang, D., Chen, J., 2018. Supervised speech separation based on deep learning: An overview. IEEE/ACM Transactions on Audio, Speech, and Language Processing 26, 1702–1726. doi:10.1109/TASLP.2018.2842159.

Wang, J., Guan, Y., Zheng, C., Peng, R., Li, X., 2021. A temporal-spectral generative adversarial network based end-to-end packet loss concealment for wideband speech transmission. The Journal of the Acoustical Society of America 150, 2577–2588. doi:10.1121/10.0006528.

Westhausen, N.L., Meyer, B.T., 2022. tPLCnet: real-time deep packet loss concealment in the time domain using a short temporal context, in: Interspeech, Incheon, South Korea. pp. 2903–2907. doi:10.21437/Interspeech.2022-10157.

Xue, H., Peng, X., Lu, Y., 2023. Contrast-plc: Contrastive learning for packet loss concealment, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece. pp. 1–5. doi:10.1109/ICASSP49357.2023.10096217.

Yamagishi, J., Veaux, C., MacDonald, K., 2019. CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92). Technical Report. University of Edinburgh. The Centre for Speech Technology Research (CSTR). doi:10.7488/ds/2645.

Yang, D.H., Chang, J.H., 2023. Towards robust packet loss concealment system with asr-guided representations, in: IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Taipei, Taiwan. pp. 1–8. doi:10.1109/ASRU57964.2023.10389616.

Yang, D.H., Chang, J.H., 2025. Flow-PLC: towards efficient packet loss concealment with flow matching. IEEE Signal Processing Letters 32, 1430–1434. doi:10.1109/LSP.2025.3553421.

Yang, D.H., Kim, D., Chang, J.H., 2023. Masked frequency modeling for improving packet loss concealment in speech transmission systems, in: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Platz, NY, USA. pp. 1–5. doi:10.1109/WASPAA58266.2023.10248056.

Záviška, P., Rajmic, P., Ozerov, A., Rencker, L., 2021. A survey and an extensive evaluation of popular audio declipping methods. IEEE Journal of Selected Topics in Signal Processing 15, 5–24. doi:10.1109/JSTSP.2020.3042071.

Zhang, G., Kleijn, W.B., 2008. Autoregressive model-based speech packet-loss concealment, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Las Vegas, NV, USA. pp. 4797–4800. doi:10.1109/ICASSP.2008.4518730.

Zhang, J., Zhao, Z., Liu, Y., Liu, J., He, Z., Niu, K., 2024a. TD-PLC: a semantic-aware speech encoding for improved packet loss concealment, in: Proc. Interspeech, Kos, Greece. pp. 1745–1749. doi:10.21437/Interspeech.2024-823.

Zhang, Z., Sun, J., Xia, X., Huang, C., Xiao, Y., Xie, L., 2024b. Bs-Plcnet: band-split packet loss concealment network with multi-task learning framework and multi-discriminators, in: IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), Seoul, South Korea. pp. 23–24. doi:10.1109/ICASSPW62465.2024.10627343.

Zhao, H., 2023. A GAN speech inpainting model for audio editing software, in: Interspeech, Dublin, Ireland. pp. 5127–5131. doi:10.21437/Interspeech.2023-904.

Zhao, Y., Bao, C., Yang, X., 2023. Predictive packet loss concealment method based on conformer and temporal convolution module, in: IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Zhengzhou, China. pp. 1–5. doi:10.1109/ICSPCC59353.2023.10400268.

Zhao, Y., Bao, C., Yang, X., Zhang, X., 2024. Time-frequency generative adversarial network with transformer for packet loss concealment, in: IEEE International Conference on Signal Processing (ICSP), Suzhou, China. pp. 404–407. doi:10.1109/ICSP62129.2024.10846185.

Zhou, Y., Bao, C., Huang, J., Zhao, Y., 2022. A neural vocoder based packet loss concealment algorithm, in: IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Xi'an, China. pp. 1–5. doi:10.1109/ICSPCC55723.2022.9984504.

Zhu, J., Bao, C., Huang, J., 2022. Packet loss concealment method based on the simplified residual network, in: IEEE International Conference on Signal Processing (ICSP), Beijing, China. pp. 51–55. doi:10.1109/ICSP56322.2022.9964486.