Débruitage de parole semi-supervisé par modélisation générative dans un espace de représentation discret des signaux audio

Sofiene KAMMOUN Simon LEGLAIVE

CentraleSupélec, IETR (UMR CNRS 6164), Avenue de la Boulaie - CS 47601, 35576 Cesson-Sévigné Cedex, France

Résumé – Le rehaussement de la parole vise à améliorer l'intelligibilité et la qualité d'un enregistrement de parole en présence de bruit. Dans cet article, nous proposons une approche semi-supervisée par modélisation générative exploitant une représentation des signaux audio sous forme de jetons discrets issus d'un codec audio neuronal. Le modèle proposé utilise l'architecture RQ-Transformer et est entraîné en deux étapes. La première étape vise à apprendre la distribution non-conditionnelle des jetons de parole propre, tandis que la seconde étape vise à affiner le modèle pour la génération de jetons de parole propre conditionnellement aux jetons de parole bruitée. Nous évaluons notre approche sur deux bases de données, permettant ainsi de mesurer la généralisation dans un domaine différent de celui d'entraînement. Les résultats expérimentaux montrent que le pré-entrainement sur des données de parole propre améliore les performances de rehaussement.

Abstract – Speech enhancement aims to improve the intelligibility and quality of a speech recording in the presence of noise. In this paper, we propose a semi-supervised approach based on generative modeling, leveraging an audio signal representation in the form of discrete tokens obtained from a neural audio codec. The proposed model uses the RQ-Transformer architecture and is trained in two stages. The first stage focuses on learning the unconditional distribution of clean speech tokens, while the second stage fine-tunes the model for generating clean speech tokens conditioned on noisy ones. We evaluate our approach on two datasets, allowing us to assess generalization in a domain different from the training one. Experimental results show that pretraining with clean speech data improves enhancement performance.

1 Introduction

Le rehaussement de la parole vise à améliorer l'intelligibilité et la qualité d'un enregistrement de parole en réduisant le bruit, la réverbération, ou toute autre forme de dégradation. Les approches traditionnelles opèrent directement sur la forme d'onde ou dans le domaine temps-fréquence, où la structure des signaux audio peut être mieux exploitée. Plus récemment, les méthodes d'apprentissage profond ont permis d'atteindre d'excellentes performances en apprenant directement une correspondance entre la parole bruitée et la parole propre. Cependant, ces modèles sont généralement entraînés de manière entièrement supervisée, s'appuyant sur de vastes bases de données de signaux de parole bruitée et propre appairés. La modélisation générative profonde constitue une alternative intéressante, car elle permet un apprentissage semi-supervisé, exploitant à la fois des données synthétiques annotées et des données réelles non annotées pour améliorer les performances et la généralisation.

Par ailleurs, les codecs audio neuronaux [14] sont aujourd'hui utilisés pour transformer un signal de parole en une représentation discrète sous forme de jetons [4]. Contrairement aux représentations continues, l'utilisation de jetons discrets simplifie les tâches de prédiction, réduit les besoins computationnels et permet d'exploiter des modèles génératifs séquentiels puissants, basés sur l'architecture Transformer, habituellement utilisés pour les modèles de langage [8]. Désignés sous le terme de modèles de langage pour la parole, ces modèles suscitent un intérêt croissant pour diverses applications telles que la synthèse de parole, la conversion de voix et la séparation de sources [2].

Plusieurs études ont exploré l'utilisation de ces modèles dans le cadre du rehaussement de la parole [5, 12, 11, 13]. Malgré leur succès, ces modèles sont entraînés de manière entièrement supervisée et nécessitent de vastes bases de données contenant des paires de signaux de parole bruitée et propre. Dans ce travail, nous proposons une approche semi-supervisée pour le rehaussement de la parole en utilisant un modèle génératif basé sur l'architecture RQ-Transformer [7] et entraîné sur les jetons de parole de FunCodec [3]. Notre méthode repose sur une phase de pré-entraînement non supervisé sur un large corpus de parole propre, permettant au modèle d'apprendre une distribution a priori des jetons de parole, avant d'être affiné pour la tâche de rehaussement. Nous montrons expérimentalement que le pré-entraînement non supervisé améliore les performances de rehaussement.

2 Méthode proposée

Le modèle proposé repose sur l'architecture RQ-Transformer [7], conçue spécifiquement pour traiter des séquences de jetons issus d'une quantification vectorielle résiduelle (RVQ). Nous proposons d'adapter de cette architecture pour le rehaussement de la parole à partir de jetons extraits d'un codec audio neuronal. Notre approche suit un processus d'entraînement en deux étapes, combinant un pré-entraînement non supervisé et un affinement supervisé. Lors du pré-entraînement, le modèle apprend à générer des jetons de parole propre à partir d'un vaste corpus de signaux de parole isolée, capturant ainsi la structure sous-jacente de la parole naturelle, comme décrit dans la section 2.2. Dans la seconde étape, le modèle est affiné de façon supervisée, afin d'adapter ses capacités génératives à la tâche de rehaussement de la parole, comme détaillé dans la section 2.3. Il est important de distinguer le codec neuronal pré-entraîné, uniquement utilisé pour extraire une représentation

discrète des signaux, et notre modèle de rehaussement, qui est pré-entraîné indépendamment pour modéliser la distribution non conditionnelle de la parole propre dans cet espace discret.

2.1 Codec audio neuronal

Soit $\bar{\mathbf{x}} \in \mathbb{R}^{d \cdot f_s}$ un signal audio de durée d en secondes et de fréquence d'échantillonnage f_s en Hertz, en supposant $d \cdot f_s \in \mathbb{N}$. Un codec audio neuronal est constitué d'un encodeur, d'un module de quantification vectorielle résiduelle (RVQ) et d'un décodeur [14].

Encodeur L'encodeur $\mathcal{E}$ transforme le signal audio $\bar{\mathbf{x}}$ en une représentation latente continue sous-échantillonnée $\bar{\mathbf{z}} = \mathcal{E}(\bar{\mathbf{x}}) \in \mathbb{R}^{L \times T}$, où $T = d \cdot f_r$ avec $f_r \ll f_s$. Cette transformation est réalisée à l'aide de couches convolutives 1D avec un pas supérieur à 1.

RVQ Le module RVQ est constitué de 2 blocs : le quantifieur $\mathcal{Q}_\mathcal{C} : \mathbb{R}^L \mapsto \{1, ..., K\}$ et son inverse $\mathcal{Q}_\mathcal{C}^{-1} : \{1, ..., K\} \mapsto \mathbb{R}^L$. Le quantifieur discrétise $\bar{\mathbf{z}}$ en utilisant un processus de quantification vectorielle (VQ) à plusieurs étages [14]. Ce module opère indépendamment sur chaque vecteur $\bar{\mathbf{z}}_t = [\bar{\mathbf{z}}]_t \in \mathbb{R}^L$, $t \in \{1, ..., T\}$. Chaque étage de quantification $n \in \{1, ..., N\}$ utilise un dictionnaire $\mathcal{C}_n = \{\mathbf{e}_k^{(n)} \in \mathbb{R}^L\}_{k=1}^K$ pour quantifier l'erreur résiduelle issue de l'étage précédent :

$$\mathbf{q}_n = \underset{\mathbf{e} \in \mathcal{C}_n}{\arg\min} \parallel \mathbf{r}_{n-1} - \mathbf{e} \parallel_2^2, \tag{1}$$

où $\mathbf{r}_0 = \bar{\mathbf{z}}_t$ et $\mathbf{r}_n = \mathbf{r}_{n-1} - \mathbf{q}_n$. Ainsi, le quantifieur $\mathcal{Q}_C$ associe le vecteur latent continu $\bar{\mathbf{z}}_t \in \mathbb{R}^L$ à un vecteur de N indices de dictionnaires, noté $\mathbf{x}_t \in \{1, ..., K\}^N$. On note $\mathbf{x} = \{\mathbf{x}_t\}_t \in \{1, \ldots, K\}^{N \times T}$ la représentation discrète obtenue pour tout le signal d'entrée. Le quantifieur inverse opère aussi indépendamment sur chaque trame temporelle afin de reconstruire les vecteurs latents continus à partir des indices : $\hat{\bar{\mathbf{z}}}_t = \mathcal{Q}_\mathcal{C}^{-1}(\mathbf{x}_t) = \sum_{n=1}^N \mathbf{q}_n$, où $\mathbf{q}_n = \mathcal{C}_n(x_{t,n})$ désigne le vecteur d'indice $x_{t,n} = [\mathbf{x}_t]_n$ dans le dictionnaire $\mathcal{C}_n$.

Décodeur Enfin, le décodeur $\mathcal{D}$ reconstruit le signal audio à partir de la représentation latente quantifiée $\hat{\bar{\mathbf{z}}} = \{\hat{\bar{\mathbf{z}}}_t\}_t \in \mathbb{R}^{L \times T}$ grâce à des couches de convolution transposée : $\hat{\bar{\mathbf{x}}} = \mathcal{D}(\hat{\bar{\mathbf{z}}}) \in \mathbb{R}^{d \cdot f_s}$.

Résumé Le fonctionnement complet du codec audio neuronal peut finalement être représenté par :

$$\bar{\mathbf{x}} \in \mathbb{R}^{d \cdot f_s} \xrightarrow{\mathcal{E}_\mathcal{C}} \mathbf{x} \in \{1, ..., K\}^{N \times T} \xrightarrow{\mathcal{D}_\mathcal{C}} \hat{\bar{\mathbf{x}}} \in \mathbb{R}^{d \cdot f_s}, \tag{2}$$

où $\mathcal{E}_\mathcal{C} = \mathcal{Q}_\mathcal{C} \circ \mathcal{E}$ et $\mathcal{D}_\mathcal{C} = \mathcal{D} \circ \mathcal{Q}_\mathcal{C}^{-1}$. Dans cet article, nous supposons le codec pré-entraîné suivant la méthodologie de [3] et développons une méthode de rehaussement de parole dans l'espace de sortie de $\mathcal{E}_\mathcal{C}$.

2.2 Modèle génératif de parole propre

Commençons par définir le modèle génératif non conditionnel des jetons de parole propre $\mathbf{x} \in \{1, ..., K\}^{N \times T}$, qui repose sur le modèle autorégressif RQ-Transformer proposé dans [7] :

$$p_\theta(\mathbf{x}) = \prod_{t=1}^{T} \prod_{n=1}^{N} p_\theta(x_{t,n} \mid \mathbf{x}_{t,1:n-1}, \mathbf{x}_{1:t-1}), \tag{3}$$

où $x_{t,n} \in \{1, ..., K\}$ est le jeton à l'instant t et à l'étage de quantification n, $\mathbf{x}_t = \{x_{t,i}\}_{i=1}^N$, $\mathbf{x}_{t,1:n} = \{x_{t,i}\}_{i=1}^n$, et $\mathbf{x}_{1:t} = \{\mathbf{x}_j\}_{j=1}^t$.

Le RQ-Transformer est un modèle qui prédit la fonction de masse de probabilité de $x_{t,n}$ étant donné $\mathbf{x}_{t,1:n-1}$ et $\mathbf{x}_{1:t-1}$. Il est composé d'un *Temporal Transformer* et d'un *Depth Transformer* permettant de modéliser séparément les dépendances temporelles et inter-étages de quantification. Ces transformeurs sont constitués de plusieurs blocs d'auto-attention masquée, chacun comprenant un module d'attention multi-tête et un réseau de neurones entièrement connecté, tous deux intégrant une connexion résiduelle et une couche de normalisation.

Le *Temporal Transformer* encode les jetons passés $\mathbf{x}_{1:t-1}$ en un unique vecteur de contexte $\mathbf{h}_t \in \mathbb{R}^H$, où H est la dimension interne du transformeur. Cette opération est réalisée comme suit. Tout d'abord, les jetons en entrée sont convertis en vecteurs $\mathcal{C}'_n(x_{t,n}) \in \mathbb{R}^L$ à l'aide d'un dictionnaire $\mathcal{C}'_n$. Bien qu'il soit possible d'apprendre de nouveaux dictionnaires, nous utilisons ceux du module de RVQ du codec audio neuronal. Si $L \neq H$, nous introduisons, pour chaque niveau de quantification, une transformation linéaire entraînable allant de $\mathbb{R}^L$ dans $\mathbb{R}^H$. Ainsi, dans la suite nous supposerons $L = H$. Ensuite, les vecteurs $\mathbf{u}_t \in \mathbb{R}^H$ sont calculés pour $t \geq 2$ comme suit :

$$\mathbf{u}_t = \mathrm{PE}_T(t) + \sum_{n=1}^{N} \left[\mathcal{C}'_n(x_{t-1,n}) + \mathrm{PE}_D(n)\right], \tag{4}$$

où $\mathrm{PE}_T(t)$ et $\mathrm{PE}_D(n)$ sont des vecteurs d'encodage positionnel pour les dimensions temporelle et de profondeur, respectivement. Pour $t = 1$, nous définissons $\mathbf{u}_1 \in \mathbb{R}^H$ comme un vecteur de paramètres entraînables. Ainsi, $\mathbf{u}_t$ encode l'ensemble des jetons à l'instant $t - 1$ et pour tous les étages de quantification. Finalement, le *Temporal Transformer* prend en entrée la séquence de vecteurs $\{\mathbf{u}_s\}_{s=1}^t$:

$$\mathbf{h}_t = \mathrm{TemporalTransformer}(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_t). \tag{5}$$

Le *Depth Transformer* encode les jetons au pas de temps actuel et aux étages de RVQ précédents $\mathbf{x}_{t,1:n-1}$, et combine cet encodage avec $\mathbf{h}_t$ pour calculer la probabilité $p_\theta(x_{t,n} = k \mid \mathbf{x}_{t,1:n-1}, \mathbf{x}_{1:t-1})$ pour tous les indices $k \in \{1, ..., K\}$. Cette opération est réalisée comme suit. Les vecteurs $\mathbf{v}_{t,i} \in \mathbb{R}^H$ sont calculés comme : $\mathbf{v}_{t,1} = \mathbf{h}_t$ et, pour $n \geq 2$,

$$\mathbf{v}_{t,n} = \mathcal{C}'_n(x_{t,n-1}) + \mathrm{PE}_D(n-1). \tag{6}$$

La séquence $\{\mathbf{v}_{t,i} \in \mathbb{R}^H\}_{i=1}^n$, qui encode donc $\mathbf{x}_{1:t-1}$ et $\mathbf{x}_{t,1:n-1}$, est utilisée par le *Depth Transformer* pour prédire un vecteur de probabilités

$$\mathbf{p}_{t,n} = \mathrm{DepthTransformer}(\mathbf{v}_{t,1}, ..., \mathbf{v}_{t,n}) \in [0, 1]^K, \tag{7}$$

avec

$$[\mathbf{p}_{t,n}]_k = p_\theta(x_{t,n} = k \mid \mathbf{x}_{t,1:n-1}, \mathbf{x}_{1:t-1}) \tag{8}$$

et $\sum_{k=1}^K [\mathbf{p}_{t,n}]_k = 1$.

Le modèle complet est entraîné de façon non supervisée, en minimisant l'erreur d'entropie croisée entre les jetons prédits et ceux vérité terrain, ce qui est équivalent à l'estimation au sens du maximum de vraisemblance pour le modèle (3).

2.3 Modèle adapté pour le rehaussement

Nous étendons dans cette section le modèle génératif précédent pour le rehaussement de la parole supervisé. L'objectif est d'apprendre la distribution conditionnelle des jetons de parole propre $\mathbf{x} \in \{1, ..., K\}^{N \times T}$ étant donné les jetons de

parole bruitée $\mathbf{y} \in \{1, ..., K\}^{N \times T}$. Nous pouvons à nouveau formuler un modèle autorégressif :

$$p_\theta(\mathbf{x} \mid \mathbf{y}) = \prod_{t=1}^{T} \prod_{n=1}^{N} p_\theta(x_{t,n} \mid \mathbf{x}_{t,1:n-1}, \mathbf{x}_{1:t-1}, \mathbf{y}). \quad (9)$$

Nous modifions la définition du vecteur de contexte dans le modèle RQ-Transformer précédent afin de prendre en compte la dépendance supplémentaire aux jetons bruités. Soit $\{\mathbf{u}'_t \in \mathbb{R}^H\}_{t=1}^T$ une séquence de vecteurs définis pour tout t par :

$$\mathbf{u}'_t = \mathrm{PE}_T(t) + \sum_{n=1}^{N} (\mathcal{C}'_n(y_{t,n}) + \mathrm{PE}_D(n)), \quad (10)$$

où $y_{t,n}$ est le jeton de parole bruitée à l'instant t et à l'étage de quantification n. Le nouveau vecteur de contexte $\mathbf{h}'_t$ est défini comme la sortie du *Temporal Transformer* prenant en entrée la concaténation des séquences $\{\mathbf{u}_s\}_{s=1}^t$ et $\{\mathbf{u}'_t\}_{t=1}^T$:

$$\mathbf{h}'_t = \mathrm{TemporalTransformer}(\mathbf{u}'_1, ..., \mathbf{u}'_T, \mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_t). \quad (11)$$

En résumé, $\mathbf{h}'_t$ encapsule l'information des jetons précédents de parole propre $\mathbf{x}_{1:t-1}$ ainsi que la séquence complète des jetons de parole bruitée $\mathbf{y}$. Le reste du modèle afin d'aboutir à la probabilité d'intérêt est inchangé par rapport au premier modèle non conditionnel, à la seule différence que $\mathbf{v}_{t,1} = \mathbf{h}'_t$.

De manière similaire à la première étape, nous entraînons le modèle en optimisant l'erreur d'entropie croisée. Lors de l'inférence, étant donné un signal de parole bruitée, nous l'encodons d'abord en une séquence de jetons discrets $\mathbf{y}$ à l'aide du codec audio neuronal. Le modèle de rehaussement de la parole traite ensuite ces jetons bruités en prédisant de manière autorégressive les jetons de parole propre. À chaque étape, le jeton prédit $x^\star_{t,n}$ est sélectionné comme suit :

$$x^\star_{t,n} = \underset{k \in \{1,...,K\}}{\mathrm{argmax}} \; p_\theta(x_{t,n} = k \mid \mathbf{x}^\star_{t,1:n-1}, \mathbf{x}^\star{}_{1:t-1}, \mathbf{y}), \quad (12)$$

où la prédiction est conditionnée par les jetons de parole propre générés précédemment ainsi que par les jetons de parole bruitée. Une fois la séquence complète de jetons propres générée, elle est décodée en un signal audio à l'aide du décodeur du codec neuronal.

3 Expériences et résultats

3.1 Configuration expérimentale

Jeux de données Pour la phase de pré-entraînement, nous utilisons le sous-ensemble `train-clean-360` du jeu de données LibriTTS [15], qui contient 191 heures de parole propre prononcée par 904 locuteurs différents. Pour la phase de rehaussement de la parole, nous utilisons deux jeux de données. Le premier, WSJ0-CHiME3 [10], est créé à partir d'énoncés de parole propre issus du *Wall Street Journal*, mélangés avec du bruit provenant de divers environnements tels que des restaurants, des carrefours routiers et des transports publics. Les valeurs de rapport signal à bruit (RSB) sont échantillonnées uniformément entre 0 et 20 dB. Ce jeu de données contient 30 heures de parole bruitée et propre pour l'entraînement, 8 heures pour la validation et 5 heures pour le test. Le second jeu de données, Libri1Mix [1], est construit à partir de livres audio et d'échantillons de bruits ambiants réels (restaurants, cafés, bars, parcs), avec des valeurs de RSB variant entre -6 et 3 dB. Nous utilisons le sous-ensemble `train-360`, qui contient 212 heures de parole bruitée et propre pour l'entraînement, 11 heures pour la validation, et 11 heures pour le test.

Configuration du modèle Pour le codec audio neuronal, nous utilisons FunCodec [3], qui propose plusieurs modèles pré-entraînés. Nous sélectionnons le modèle LibriTTS avec une fréquence d'échantillonnage de $f_r = 25$ Hz dans l'espace latent. Ce modèle dispose de 32 niveaux de quantification, chacun associé à un dictionnaire contenant $K = 1024$ vecteurs de dimension $L = 128$. Pour nos expériences, nous utilisons uniquement les $N = 7$ premiers niveaux de quantification, ce qui réduit la longueur de la séquence d'entrée d'un facteur 4.5 sans dégrader significativement la qualité audio.

Pour le modèle RQ-Transformer, nous utilisons 5 couches pour le *Temporal Transformer* et 10 pour le *Depth Transformer*. Chaque couche comprend un module d'attention multi-têtes avec 8 têtes d'attention et un réseau entièrement connecté avec une dimension cachée de $H = 512$. Nous utilisons les dictionnaires du modèle FunCodec, avec pour chaque étage de quantification une couche convolutive 1D entraînable de taille de noyau 1 et avec $H = 512$ canaux de sortie.

Entraînement Nous entraînons nos modèles en échantillonnant aléatoirement des séquences de 2 secondes à partir de l'ensemble d'entraînement et en extrayant les jetons à l'aide du codec audio neuronal. Cela correspond à $7 \times 25 \times 2$ jetons par exemple d'entraînement. Avant l'extraction des jetons, nous normalisons les signaux audio pour que leur moyenne quadratique soit égale à 1. Afin d'analyser l'effet du pré-entraînement, nous entraînons le même modèle deux fois, avec et sans pré-entraînement. De plus, nous entraînons des modèles distincts sur chaque jeu de données annoté afin d'évaluer la performance inter-jeux de données et la capacité de généralisation.

Nous entraînons nos modèles à l'aide de deux GPU NVIDIA HGX A100 pour le pré-entraînement et l'affinage. Nous pré-entraînons d'abord un modèle RQ-Transformer sur le jeu de données de parole propre pendant 50 époques, puis nous affinons chaque modèle pendant 300 époques sur le jeu de données annoté avec une taille de lot de $B = 60$ par GPU. Nous utilisons l'optimiseur AdamW avec $\beta_1 = 0.9$, $\beta_2 = 0.95$, et une régularisation $L2$ avec un poids de 0.05, ainsi qu'un planificateur du taux d'apprentissage à décroissance cosinus, avec un échauffement sur 10 époques. Le taux d'apprentissage de base est fixé à $0.001 \times (B \div 256)$, conformément aux stratégies courantes.

Métriques Pour évaluer les performances de notre modèle, nous nous appuyons sur des métriques de qualité non-intrusives, couramment utilisées dans la littérature pour les modèles génératifs de rehaussement de la parole [11, 12, 13]. Ce choix est motivé par le fait que les codecs audio neuronaux apprennent à reconstruire la phase des signaux au travers d'un entraînement adversaire et non d'une fonction de coût mesurant la reconstruction à l'échelle des échantillons de la forme d'onde. Les signaux reconstruits ne sont donc pas parfaitement alignés avec les signaux de référence, ce qui dégrade fortement les métriques standards telles que le rapport signal à bruit, sans pour autant nuire à la qualité de reconstruction perçue [6]. Ainsi, des métriques comme DNSMOS, conçues pour prédire les évaluations humaines, sont devenues la norme pour

TABLE 1 : Résultats de rehaussement sur les bases de test WSJ0-CHiME3 et Libri1Mix. *Les modèles M1 et M2 sont entraînés sur Libri1Mix et WSJ0-CHiME3, respectivement. †PNS indique "Pré-entraînement non supervisé".

		Libri1Mix			WSJ0-CHiME3		
Modèle*	PNS†	OVRL	SIG	BAK	OVRL	SIG	BAK
M1	✗	2.83	**3.42**	3.36	3.02	**3.50**	3.63
	✓	**2.89**	3.41	**3.49**	**3.06**	**3.50**	**3.70**
M2	✗	2.73	3.37	3.22	2.92	3.43	3.50
	✓	2.81	3.39	3.37	2.96	3.44	3.57
Bruité	-	1.75	2.46	1.81	2.74	3.46	3.15

l'évaluation du rehaussement de la parole dans ce contexte. Plus précisément, nous utilisons le DNSMOS P.835 [9], une métrique non-intrusive entraînée pour prédire les évaluations humaines en débruitage. DNSMOS évalue le rehaussement de la parole selon trois dimensions : SIG (qualité de la parole), BAK (suppression du bruit) et OVRL (qualité globale), avec un système de score allant de 1 (mauvais) à 5 (excellent).

3.2 Résultats

Le Tableau 1 présente les résultats de rehaussement de la parole pour les modèles entraînés sur Libri1Mix (modèle M1) et WSJ0-CHiME3 (modèle M2). Chaque modèle est évalué à la fois sur son ensemble de test correspondant et sur un ensemble de test différent afin d'évaluer la généralisation entre jeux de données. Le tableau inclut également les résultats des modèles pré-entraînés (PNS = ✓) et non pré-entraînés (PNS = ✗), ce qui nous permet d'analyser l'impact du pré-entraînement.

Pour toutes les métriques sauf SIG sur la base Libri1Mix, les modèles pré-entraînés surpassent leurs homologues non pré-entraînés. Cette amélioration se manifeste aussi bien dans des conditions de test similaires à l'entraînement que dans des conditions différentes, avec un effet encore plus prononcé dans ces dernières. Par exemple, pour le modèle M2 on observe une amélioration de 0.07 en BAK et 0.04 en OVRL dans des conditions de test similaires à l'entraînement, tandis que dans des conditions différentes, ces gains atteignent 0.15 en BAK et 0.08 en OVRL. Cela suggère que le pré-entraînement améliore la généralisation à des conditions différentes de celles d'entraînement. Par ailleurs, tous les modèles obtiennent de bonnes performances sur la métrique BAK, indiquant une suppression efficace du bruit de fond. Ce résultat est attendu, car ces modèles sont entraînés pour générer des jetons de parole propre, donc sans allouer de masse de probabilité aux jetons représentant du bruit.

4 Conclusion

Dans cet article, nous avons introduit une approche générative pour le rehaussement de la parole utilisant une architecture RQ-Transformer [7] entraînée sur les jetons de parole issus de FunCodec [3]. Notre méthode suit un processus en deux étapes : un pré-entraînement non supervisé sur de la parole propre, suivi d'un affinement pour le rehaussement de la parole. Nos résultats mettent en évidence les bénéfices du pré-entraînement, conduisant à une amélioration des performances sur les métriques de qualité non intrusives. Les travaux futurs pourraient explorer d'autres mécanismes de conditionnement, des techniques d'échantillonnage améliorées et l'utilisation de codecs audio neuronaux plus performants.

Remerciements Ce travail a été réalisé en utilisant les ressources de calcul du Mésocentre de l'Université Paris-Saclay, CentraleSupélec et de l'École Normale Supérieure Paris-Saclay, dans le cadre du projet DEGREASE (ANR-23-CE23-0009), financé par l'Agence Nationale de la Recherche.

Références

[1] J. COSENTINO *et al.* : LibriMix : An open-source dataset for generalizable speech separation. *arXiv :2005.11262*, 2020.

[2] W. CUI *et al.* : Recent advances in speech language models : A survey. *arXiv :2410.03751*, 2024.

[3] Z. DU *et al.* : FunCodec : A fundamental, reproducible and integrable open-source toolkit for neural speech codec. *In Proc. IEEE ICASSP*, 2024.

[4] Y. GUO *et al.* : Recent advances in discrete speech tokens : A review. *arXiv :2502.06490*, 2025.

[5] B. KANG *et al.* : LLaSE-G1 : Incentivizing generalization capability for LLaMA-based speech enhancement. *arXiv :2503.00493*, 2025.

[6] R. KUMAR *et al.* : NU-GAN : High resolution neural upsampling with gan. *arXiv :2010.11362*, 2020.

[7] D. LEE *et al.* : Autoregressive image generation using residual quantization. *In Proc. CVPR*, 2022.

[8] J. LI *et al.* : Investigating neural audio codecs for speech language model-based speech generation. *In Proc. SLT*, 2024.

[9] C. K. REDDY *et al.* : DNSMOS : A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. *In Proc. IEEE ICASSP*, 2021.

[10] J. RICHTER *et al.* : Speech enhancement and dereverberation with diffusion-based generative models. *IEEE/ACM Trans. ASLP*, 31, 2023.

[11] Z. WANG *et al.* : SELM : Speech enhancement using discrete tokens and language models. *In Proc. IEEE ICASSP*, 2024.

[12] H. YANG *et al.* : Genhancer : High-fidelity speech enhancement via generative modeling on discrete codec tokens. *In Proc. Interspeech*, 2024.

[13] J. YAO *et al.* : GenSE : Generative speech enhancement via language models using hierarchical modeling. *In Proc. ICLR*, 2025.

[14] N. ZEGHIDOUR *et al.* : SoundStream : An end-to-end neural audio codec. *IEEE/ACM Trans. ASLP*, 30, 2022.

[15] H. ZEN *et al.* : LibriTTS : A corpus derived from librispeech for text-to-speech. *In Proc. Interspeech*, 2019.