Quantification optimisée de l'espace latent en codage audio neuronal

Thomas MULLER[1, 2] Stéphane RAGOT[1] Quentin LEMESLE[1] Pierrick PHILIPPE[1] Pascal SCALART[2]

[1]Orange Labs, 2 Avenue Pierre Marzin, 22300 Lannion, France

[2]Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), 6 Rue de Kerampont, 22300 Lannion, France

Résumé – Cet article se concentre sur le codage audio par réseaux de neurones artificiels. Nous proposons d'appliquer une analyse et une transformation de l'espace latent par décomposition en valeurs propres, afin de modifier voire remplacer la quantification vectorielle résiduelle (RVQ) actuellement utilisée par des codecs récents tels que SoundStream ou EnCodec. L'approche proposée permet en particulier une réduction du stockage et de la complexité d'environ 37% pour EnCodec sans dégrader la qualité audio.

Abstract – This article focuses on audio coding based on artificial neural networks. We propose to analyze and transform the latent space based on an eigenvalue decomposition, in order to modify or even replace the residual vector quantization (RVQ) used in recent codecs such as SoundStream and EnCodec. In particular, the proposed approach brings about 37% of reduction in storage and computational complexity for EnCodec, with no quality degradation.

1 Introduction

Le codage audio avec perte est au coeur des modes de communication numériques, par exemple pour le transport de la voix en téléphonie ou la diffusion de musique sur Internet. Réalisé via un codec (**co**deur-**dec**odeur), le codage d'une source audio peut être divisé en trois fonctions : l'analyse, la quantification – éventuellement suivie par un codage entropique – et la synthèse (voir figure 1). L'analyse consiste à représenter le signal dans un autre domaine : par exemple le transformer de façon à concentrer son énergie sur peu de coefficients, ou le décrire avec un modèle paramétrique. La quantification est optimisée pour atteindre le meilleur compromis débit/distorsion. Enfin, la synthèse ramène la représentation quantifiée dans le domaine du signal. Les codecs audio conventionnels comme Opus [1] et EVS [2] utilisent pour l'analyse et la synthèse des méthodes issues du traitement du signal conventionnel, notamment la prédiction linéaire et le codage par transformée.

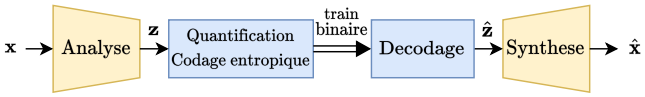

FIGURE 1 : Schéma général d'un codec.

Ces dernières années, comme dans de nombreux autres domaines, le paradigme de l'apprentissage automatique à partir de données a bouleversé l'état de l'art. De nouvelles méthodes de représentations neuronales ont fait leur apparition sous l'impulsion de WaveNet [3] pour la génération de signaux audio et du modèle VQ-VAE [4] pour intégrer la quantification d'un espace latent dès l'entraînement du réseau. De nombreuses approches de codage de parole et audio ont été explorées avec par exemple Lyra [5], LPCNet [6], ou MelGAN [7]. Une nouvelle génération de codecs audio s'est développée à partir de ces architectures de réseaux de neurones, dont SoundStream (Lyra V2) [8] et EnCodec [9]. Ces deux codecs atteignent des débits très faibles de l'ordre de quelques kilobits par seconde (kbps), ils sont plus complexes que les codecs conventionnels conçus à partir de connaissances expertes sur le signal, mais ils fonctionnent en temps-réel, et l'entraînement sur des données réelles permet d'atteindre une reconstruction fidèle de l'audio.

SoundStream et EnCodec utilisent une quantification vectorielle résiduelle (RVQ), aussi appelée quantification multi-étages [10], dont les dictionnaires sont appris lors de l'entraînement bout-en-bout du modèle, permettant d'apprendre conjointement les mots de code avec l'ensemble des poids du réseau de neurones. Cependant, comme toute approche d'apprentissage, la RVQ requiert de stocker les dictionnaires de chaque étage de quantification, et la recherche du plus proche voisin est relativement complexe. Ainsi, l'objectif de la présente étude est d'optimiser la quantification de l'espace latent afin de diminuer le stockage et la complexité, et d'explorer comment en améliorer les performances. L'étude s'est concentrée sur EnCodec qui dispose d'un code Python public avec les poids du réseau entraîné. Les contributions de cet article sont les suivantes :

- Analyse de l'espace latent et transformation de cet espace par décomposition en valeurs propres avant quantification.
- Modification des dictionnaires de quantification pour les tronquer et réduire le stockage et la complexité de la RVQ.

Cet article est organisé comme suit. La section 2 résume les grands principes de la compression audio utilisant le modèle neuronal EnCodec, la section 3 illustre notre proposition d'optimisation de l'étage de quantification. Les résultats expérimentaux sont exposés dans la section 4 avant d'explorer la possibilité de remplacer la quantification RVQ à la section 5 et de conclure à la section 6.

2 Compression audio avec EnCodec

EnCodec [9] est un codec audio basé sur un réseau de neurones divisé en deux parties : l'encodeur $\mathcal{E}$ pour l'analyse et le décodeur $\mathcal{D}$ pour la synthèse. L'architecture du réseau est symétrique, avec plusieurs couches de convolution 1D et des connexions résiduelles facilitant la rétropropagation du gradient lors de la phase d'apprentissage. Des couches de sous-échantillonnage sont intégrées sous forme de convolution à décimation ("strided"). Afin d'avoir un champ récepteur

plus large dans le temps, une partie des convolutions utilise de la dilatation [11]. Des couches de neurones récursifs sont également ajoutées proche de l'étage de quantification pour modéliser la dimension temporelle des données. L'architecture de l'encodeur est telle que 320 échantillons d'audio échantillonné à 24 kHz sont transformés en un vecteur de l'espace latent de dimension 128.

On s'intéresse dans cette étude à la quantification vectorielle résiduelle (RVQ) qui sépare l'encodeur et le décodeur. Son fonctionnement est illustré à la figure 2. Elle consiste en la mise en cascade d'étages de quantification, chacun quantifiant le résidu (l'erreur) de quantification de l'étage précédent. Ainsi, plus il y a d'étages de quantification, plus l'erreur globale de quantification est faible, mais plus le débit est important. La recherche du meilleur mot de code pour quantifier un vecteur est réalisée en minimisant la distance euclidienne entre le vecteur à coder et les mots de code présents dans le dictionnaire appris lors de l'entraînement. Le vecteur latent (de dimension 128) est quantifié avec un budget binaire allant de 20 à 320 bits, ce qui correspond à un débit allant de 1,5 à 24 kbps. Le budget binaire utilisé est donné par le nombre de quantificateurs vectoriels Q_i utilisés. Chaque étage utilise un dictionnaire $\mathcal{C}_i$ de 1024 vecteurs de dimension 128 indexés sur 10 bits. EnCodec fonctionne avec 2, 4, 8, 16 ou 32 étages.

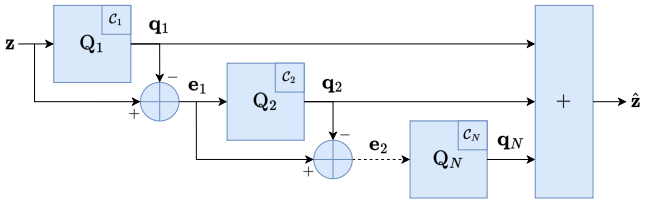

FIGURE 2 : Quantification vectorielle résiduelle. Mise en cascade de N quantificateurs vectoriels Q_i, chacun codant le résidu de quantification de l'étage précédent. La reconstruction du vecteur correspond à la somme des mots décodés.

Le modèle est entraîné de bout-en-bout avec plusieurs fonctions de coût. Une partie se concentre sur la fidélité de la reconstruction, dans le domaine temporel et fréquentiel, et à plusieurs échelles. L'autre partie correspond à un entraînement de réseau antagoniste génératif avec l'aide d'un discriminateur, permettant de produire des signaux audio les plus réalistes possible. Le dictionnaire du quantificateur est entraîné selon la méthode proposée dans VQ-VAE [4].

Bien qu'EnCodec soit également capable de traiter de l'audio stéréo et intègre un modèle de langage permettant d'améliorer la compression des données binaires, nous nous restreignons à son fonctionnement sur des signaux mono échantillonnés à 24 kHz. Dans notre étude, nous utilisons la version déjà entraînée du modèle, en ne modifiant que la quantification.

3 Quantification RVQ optimisée

La méthode proposée a pour but de réduire le stockage et la complexité de la recherche du plus proche voisin. Elle s'appuie sur l'observation qu'il existe une corrélation importante entre les composantes des vecteurs latents issus de l'encodeur. Il est ainsi possible de décorréler ces données pour les quantifier plus efficacement.

La nouvelle chaîne de quantification proposée est illustrée figure 3. Soit un signal audio normalisé $\mathbf{x} \in [-1, 1]^L$ avec L le nombre d'échantillons. Sans perte de généralité, nous fixons $L = 320$ ce qui correspond à la taille d'une trame de signal traitée selon les paramètres de l'architecture du modèle neuronal. L'encodeur $\mathcal{E}$ génère une représentation latente $\mathbf{z} = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{128}$. Nous utilisons la transformation de Karhunen-Loeve (KLT) [12] sur le vecteur centré $\mathbf{z}' = \mathbf{z} - \boldsymbol{\mu}$ afin d'avoir un vecteur transformé $\mathbf{y}$ ayant des composantes décorrélées. Cela consiste à diagonaliser la matrice de corrélation de la représentation latente centrée $\mathbf{R} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$ et utiliser la matrice des vecteurs propres $\mathbf{U}$ pour transformer la représentation latente centrée (rotation $\mathbf{y} = \mathbf{U}^{\mathrm{T}}\mathbf{z}'$). Il est alors possible d'observer les valeurs propres de cette décomposition contenues dans $\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_i)$ en les triant par valeur décroissante, comme montré à la figure 4. On peut voir qu'il y a une forte diminution de l'amplitude autour de la 75ème valeur propre. Cela nous a poussé à tronquer la représentation latente transformée à $M = 80$ dimensions au lieu de 128, amenant à un gain de stockage et une réduction de complexité. L'équation (1) résume ces étapes précédant la quantification, où $\mathrm{Tronc}_M[\cdot]$ correspond à l'opérateur de troncature d'un vecteur à ses M premières composantes

$$\mathbf{y}_{\mathrm{T}} = \mathrm{Tronc}_M\left[\mathbf{U}^{\mathrm{T}}(\mathbf{z} - \boldsymbol{\mu})\right]. \tag{1}$$

Le vecteur transformé $\mathbf{y}$ est ainsi tronqué à 80 dimensions en $\mathbf{y_T}$ et est quantifié ($\hat{\mathbf{y}}_{\mathbf{T}} = \mathrm{Q}(\mathbf{y_T})$). La transformation inverse est appliquée, à savoir extension du vecteur avec des zéros pour les dimensions tronquées, rotation inverse $\mathbf{U}\hat{\mathbf{y}}$ et ajout de la moyenne $\boldsymbol{\mu}$. Ces étapes sont résumées par l'équation (2), avec $\mathrm{Ext}_P[\cdot]$ l'opérateur d'extension d'un vecteur de P composantes nulles

$$\hat{\mathbf{z}} = \mathbf{U}\,\mathrm{Ext}_P\left[\hat{\mathbf{y}}_{\mathrm{T}}\right] + \boldsymbol{\mu}\,. \tag{2}$$

Finalement, le décodeur $\mathcal{D}$ reconstruit l'audio $\hat{\mathbf{x}} = \mathcal{D}(\hat{\mathbf{z}})$.

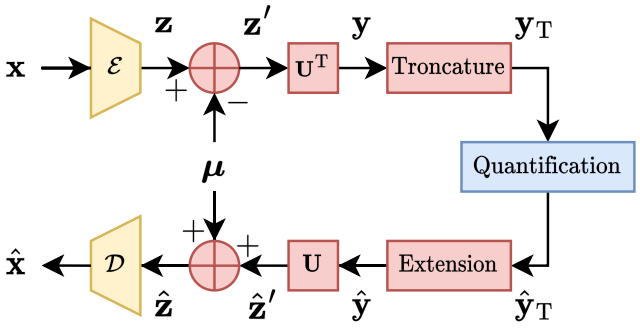

FIGURE 3 : Quantification modifiée de l'espace latent.

Cependant, le quantificateur vectoriel résiduel d'EnCodec est entraîné pour quantifier des vecteurs de dimension 128 et n'ayant pas subi cette transformation KLT. La méthode d'optimisation de la quantification consiste alors à appliquer cette même transformation aux dictionnaires et à les tronquer. Ainsi, tous les mots de code subissent la rotation décrite par la matrice des vecteur propres $\mathbf{U}$ et sont tronqués à 80 dimensions pour donner les dictionnaires modifiés $\tilde{\mathcal{C}}_i$. Afin de rester cohérent avec le centrage des données, la même moyenne $\boldsymbol{\mu}$ est retranchée aux mots de code du premier dictionnaire et au vecteur latent. La modification des dictionnaires est résumée par l'équation (3).

$$\begin{cases} \tilde{\mathcal{C}}_1 = \mathrm{Tronc}_M\left[\mathbf{U}^{\mathrm{T}}(\mathcal{C}_1 - \boldsymbol{\mu})\right] \\ \tilde{\mathcal{C}}_k = \mathrm{Tronc}_M\left[\mathbf{U}^{\mathrm{T}}\mathcal{C}_k\right] \quad k = 2, \ldots, 32 \end{cases} \tag{3}$$

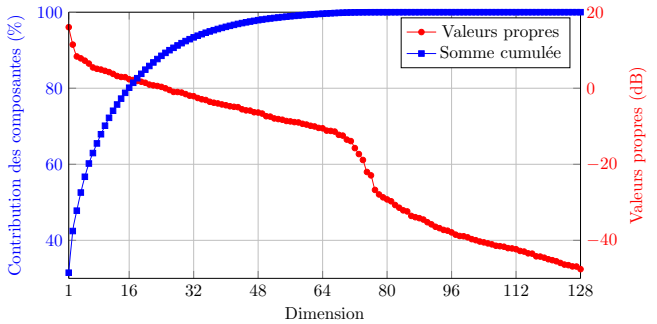

FIGURE 4 : Valeurs propres λ_i (triées) issues de la diagonalisation de la matrice de corrélation $\mathbf{R}$. En rouge : valeur propre pour chaque dimension (de 1 à 128), échelle en dB. En bleu : somme cumulée des valeurs propres, affichée en pourcentage de la somme totale.

La matrice de corrélation $\mathbf{R}$ utilisée pour calculer $\mathbf{U}$ est obtenue à partir des premiers dictionnaires RVQ. Plutôt que de calculer $\mathbf{R}$ uniquement avec les 1024 mots de code du premier dictionnaire, nous l'avons calculé sur les 1024^2 mots de code obtenus en combinant les deux premiers dictionnaires, à savoir toutes les combinaisons $c_1^i + c_2^j$ où c_k^i est le $i$ème mot de code du dictionnaire $\mathcal{C}_k$.

Pour la moyenne $\boldsymbol{\mu}$, nous avions initialement utilisé la moyenne du premier dictionnaire mais celle-ci introduit un biais sur la synthèse des trames de silence qui se traduit par une dégradation de la qualité audio. Finalement nous utilisons la moyenne de trames de silence provenant d'une base de données interne.

Il est important de noter que la transformation est fixe (non adaptative), la moyenne $\boldsymbol{\mu}$, la matrice des vecteurs propres $\mathbf{U}$ et les dictionnaires modifiés $\tilde{\mathcal{C}}_i$ sont pré-calculés et stockés. Ainsi, la modification proposée ne demande aucune transmission d'information supplémentaire.

4 Évaluation de la quantification RVQ modifiée dans EnCodec

Afin de valider la méthode proposée, nous avons effectué des tests objectifs et subjectifs. Nous comparons la version originale d'EnCodec à plusieurs débits binaires (1,5 ; 3 ; 6 ; 12 et 24 kbps) ainsi que ce même codec utilisant l'optimisation du dictionnaire de quantification décrite précédemment.

Plusieurs métriques objectives sont utilisées pour diversifier les évaluations automatiques de la qualité audio : STOI [13], ViSQOL [14] et WARP-Q [15]. Pour ces trois métriques, les scores donnés par ces algorithmes sont respectivement un score d'intelligibilité entre 0 et 100 %, un score appelé MOS-LQO entre 1 et 5 et une note MOS prédite entre 1 et 5, la qualité de l'audio estimée étant d'autant meilleure que les scores sont élevés. Ces tests ont été effectués sur environ 25 min d'audio issu de la base de données VCTK [16]. Les scores de ces trois métriques sont consignés table 1. Pour chaque débit, nous présentons le score d'EnCodec (référence Ref) et la différence des scores Δ = score(EnCodec modifié) − score(EnCodec original). Les différences de score Δ sont négatives ce qui est normal car la troncature enlève de l'information et ne peut que dégrader la qualité de la quantification ; mais les valeurs Δ très proches de zéro suggèrent une dégradation peu ou pas perceptible.

L'oreille humaine restant le meilleur outil pour juger la qualité d'un signal audio, une étude subjective informelle a été réalisée aux différents débits sous la forme de tests AB de comparaison par paires sur une échelle discrète de qualité sur 7 points de -3 à 3 (3/-3 = A/B bien meilleur, 2/-2 = A/B meilleur, 1/-1 = A/B légèrement meilleur, 0 = A et B identiques). Elle a été effectuée sur dix fichiers de parole de la base VCTK et dix fichiers d'une base audio interne sur lesquels les différences objectives étaient respectivement les plus fortes.

Ces tests subjectifs, réalisés par trois sujets experts sur les 20 échantillons les plus critiques, indiquent que globalement à 1,5, 3 et 6 kbps les différences ne sont pas statistiquement significatives, tandis qu'à 12 et 24 kbps EnCodec original et EnCodec modifié sont statistiquement différents mais avec une différence globale très faible (en moyenne 0,1) qui reste proche de la note 0 ("A et B identiques").

TABLE 1 : Métriques objectives pour EnCodec original (Ref) et différences (Δ) des scores entre EnCodec original et EnCodec modifié comme à la figure 3 (moyenne $\pm$ écart-type).

Débit (kbps)	– –	**STOI** (Intelligibilité)	**ViSQOL** (MOS-LQO)	**WARP-Q** (MOS prédit)
1,5	Ref	$73,9 \pm 10,3$	$2,47 \pm 0,50$	$2,56 \pm 0,32$
	Δ	$-0,20 \pm 0,43$	$-0,06 \pm 0,11$	$-0,01 \pm 0,13$
3	Ref	$78,2 \pm 10,5$	$3,09 \pm 0,54$	$2,82 \pm 0,33$
	Δ	$-0,19 \pm 0,37$	$-0,05 \pm 0,11$	$0,00 \pm 0,13$
6	Ref	$82,1 \pm 10,3$	$3,45 \pm 0,54$	$3,05 \pm 0,31$
	Δ	$-0,20 \pm 0,40$	$-0,04 \pm 0,11$	$-0,01 \pm 0,13$
12	Ref	$85,1 \pm 10,0$	$3,63 \pm 0,54$	$3,21 \pm 0,28$
	Δ	$-0,22 \pm 0,43$	$-0,05 \pm 0,11$	$-0,02 \pm 0,11$
24	Ref	$86,9 \pm 9,7$	$3,70 \pm 0,50$	$3,30 \pm 0,27$
	Δ	$-0,22 \pm 0,38$	$-0,04 \pm 0,09$	$-0,01 \pm 0,08$

Concernant le gain de stockage, les 32 dictionnaires $\mathcal{C}_k$ sont représentés par $32 \times 128 \times 1024$ nombres flottants. La troncature des dictionnaires à 80 dimensions au lieu de 128 permet de diminuer ce nombre à $32 \times 80 \times 1024$, en notant qu'il faut cependant stocker la moyenne $\boldsymbol{\mu}$ et la matrice des vecteurs propres $\mathbf{U}$, soit $128 + 128 \times 128$ nombres flottants. Ainsi, notre méthode permet un gain de stockage de $37,1\%$.

Pour ce qui est de la complexité, la méthode rajoute deux additions ($\boldsymbol{\mu}$ et $-\boldsymbol{\mu}$), deux multiplications matricielles ($\mathbf{U}$ et $\mathbf{U}^{\mathrm{T}}$) et la troncature et l'extension de vecteurs. En revanche, elle permet de réduire la complexité de la recherche du plus proche voisin en comparant des vecteurs de dimension 80 et non plus 128.

5 Discussion et suite des travaux

La motivation initiale de cette étude portait sur la possibilité de changer la RVQ par un autre type de quantification. En effet, d'une part la RVQ peut être complexe et nécessiter un important stockage des dictionnaires, et d'autre part elle n'est pas optimale à haut débit avec l'utilisation de nombreux étages de quantification. Pour identifier les méthodes utilisables en remplacement de la RVQ d'EnCodec, il nous a fallu étudier la nature de l'espace latent, et en particulier de l'espace latent transformé après centrage et rotation. La figure 5 montre

les 3 premières composantes de l'espace latent transformé $\mathbf{y} = (y_1 y_2 \dots y_{128})$ au cours du temps, en regard du signal de parole d'entrée, quand les paramètres $\boldsymbol{\mu}$ et $\mathbf{U}$ sont déterminés comme à la section 3. Nous observons que la première composante y_1 semble correspondre au niveau sonore du signal d'entrée (au signe près). En revanche, les autres composantes restent difficiles à analyser.

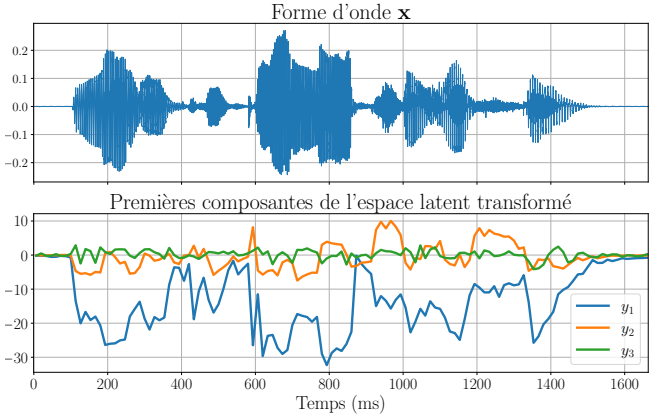

FIGURE 5 : Analyse des premières composantes de l'espace latent modifié.

Une autre approche a été de tracer les histogrammes des différentes composantes de l'espace latent modifié. Les distributions des trois premières dimensions sont présentées à la figure 6 en rouge. Autant à partir de la troisième composante les distributions semblent laplaciennes avec un pic supplémentaire en zéro dû au silence du signal de parole, autant ce n'est pas le cas pour les deux premières dimensions. Nous avons constaté que la forme de la distribution des premières composantes semblait venir de la non-stationnarité du signal d'entrée, et principalement des fortes différences statistiques entre de la parole active et du silence. Ne centrant qu'avec une seule moyenne globale $\boldsymbol{\mu}$, l'hypothèse d'un vecteur centré n'est donc plus vérifiée à court terme. Une moyenne adaptative serait plus appropriée, ce qui peut être obtenu en utilisant le premier étage de quantification Q_1 de la RVQ. La distribution de l'erreur de quantification $\mathbf{e}_1$ après quantification de $\mathbf{z}$ par Q_1 correspond à la courbe bleue sur la figure 6. Cette modification permet d'avoir des distributions beaucoup plus semblables et avec une allure laplacienne, ce qui permettrait de mieux conditionner les vecteurs en entrée d'une quantification vectorielle algébrique. Des investigations préliminaires ont été menées, dans lesquelles l'espace latent $\mathbf{z}$ est transformé comme à la figure 3 et où la quantification du signal transformée $\mathbf{y}_\mathrm{T}$ est réalisée selon la méthode décrite dans [17].

6 Conclusion

Nous avons proposé une méthode permettant un gain de stockage et de complexité de la quantification RVQ, sans dégradation de la qualité audio, ni débit supplémentaire, pour l'exemple du codec audio neuronal EnCodec. Bien que la méthode proposée ait été appliquée sur les dictionnaires appris pour EnCodec, elle est assez générale pour être appliquée à d'autres codecs utilisant une quantification vectorielle résiduelle. Une investigation de l'espace latent transformé laisse penser qu'il serait possible de remplacer la quantification RVQ par une alternative plus performante.

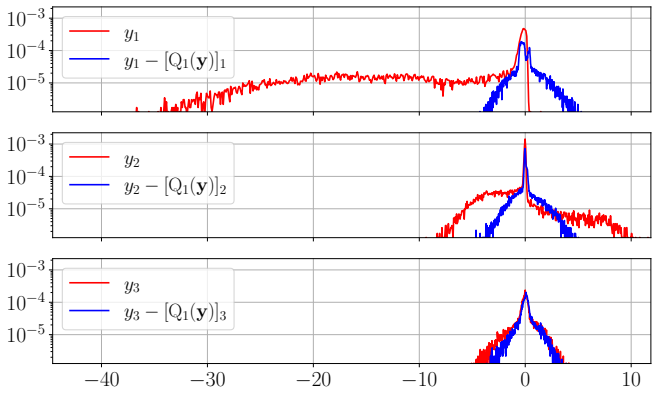

FIGURE 6 : Histogrammes des trois premières composantes de l'espace latent modifié, avant et après quantification par Q_1.

Références

[1] J.-M. Valin, K. Vos, and T. B. Terriberry, "Definition of the Opus Audio Codec." RFC 6716, Sept. 2012.

[2] M. Dietz *et al.*, "Overview of the EVS codec architecture," in *Proc. ICASSP*, pp. 5698–5702, 2015.

[3] A. van den Oord *et al.*, "WaveNet : A Generative Model for Raw Audio," in *arXiv :1609.03499*, 2016.

[4] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," in *Proc. NeurIPS*, 2018.

[5] W. B. Kleijn *et al.*, "Generative speech coding with predictive variance regularization," in *Proc. ICASSP*, 2021.

[6] J.-M. Valin and J. Skoglund, "LPCNET : Improving Neural Speech Synthesis through Linear Prediction," in *Proc. ICASSP*, pp. 5891–5895, 2019.

[7] K. Kumar and al., "MelGAN : Generative Adversarial Networks for Conditional Waveform Synthesis," in *Proc. NeurIPS*, 2019.

[8] N. Zeghidour *et al.*, "SoundStream : An End-to-End Neural Audio Codec," *IEEE/ACM Trans. TASLP*, 2021.

[9] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High Fidelity Neural Audio Compression," in *arXiv :2210.13438*, 2022.

[10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1991.

[11] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," in *arXiv :1603.07285*, 2018.

[12] N. Jayant and P. Noll, *Digital Coding of Waveforms : Principles and Applications to Speech and Video*. Prentice-Hall, 1984.

[13] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. ICASSP*, pp. 4214–4217, 2010.

[14] M. Chinen *et al.*, "ViSQOL v3 : An Open Source Production Ready Objective Speech and Audio Metric," in *Proc. QoMEX*, 2020.

[15] W. A. Jassim *et al.*, "WARP-Q : Quality Prediction For Generative Neural Speech Codecs," in *Proc. ICASSP*, 2021.

[16] J. Yamagishi *et al.*, "CSTR VCTK Corpus : english multi-speaker corpus for CSTR voice cloning toolkit," 2019.

[17] S. Ragot *et al.*, "Low-complexity multi-rate lattice vector quantization with application to wideband TCX speech coding at 32 kbit/s," in *Proc. ICASSP*, vol. 1, pp. I–501, 2004.