Approche Hybride pour l'Inversion de la Compression Dynamique en Traitement Audio à l'aide de l'Apprentissage Profond

Haoran SUN Dominique FOURER Hichem MAAREF

IBISC (EA4526), Univ. Evry - Paris-Saclay
Évry-Courcouronnes, France

Résumé – La compression de la dynamique du signal (abr. angl. DRC) est un effet audio non linéaire couramment utilisé en studio et en production musicale. Inverser cet effet est essentiel pour restaurer la dynamique originale d'un signal ainsi que sa qualité audio. Dans ce travail, nous proposons une approche hybride combinant un modèle paramétrique d'inversion de la compression dynamique avec des réseaux de neurones profonds. Cette combinaison permet à la fois une estimation robuste des paramètres du compresseur et une restauration plus fidèle du signal audio. Notre méthode repose sur deux architectures neuronales : l'une pour estimer la classe du compresseur et l'autre pour estimer directement ses paramètres. Les évaluations comparatives menées sur plusieurs jeux de données montrent une amélioration significative de la restauration audio par rapport aux méthodes de l'état de l'art.

Abstract – Dynamic range compression (DRC) is a nonlinear audio effect commonly used in studios and music production. Reversing this effect is essential for restoring a signal's original dynamic range and audio quality. In this work, we propose a hybrid approach combining a parametric dynamic range compression inversion model with deep neural networks. This combination enables both robust estimation of compressor parameters and more faithful restoration of the audio signal. Our method relies on two neural architectures: one to estimate the compressor class and the other to directly estimate its parameters. Comparative evaluations conducted on several datasets show significant improvements in audio restoration when compared to some state-of-the-art methods.

1 Introduction

La Compression Dynamique (DRC) est une technique de traitement audio permettant de changer la dynamique d'un signal. Elle est largement utilisée dans l'enregistrement, le mixage et la mastérisation [1]. Bien que la DRC soit essentielle, elle modifie le timbre et la qualité audio perçue, et son inversion reste un défi pour des applications telles que la restauration audio ou la rétro-ingénierie des mixages [2]. Les méthodes d'inversion de la DRC existantes sont confrontées à des limitations. Dans [3], les auteurs proposent d'utiliser une approche informée combinant estimation et codage. Plus tard, [4] introduit un modèle paramétrique permettant une meilleure restauration du signal lorsque les paramètres du compresseur sont exactement connus. Les méthodes actuelles basées sur l'apprentissage profond [5, 6] fournissent des résultats convaincants mais ne fonctionnent que pour des types de DRC spécifiques correspondant aux données disponibles durant l'apprentissage.

Nous introduisons ici une approche hybride reposant sur l'apprentissage profond et une approche modèle qui fonctionne en 2 étapes : 1) un réseau de neurones profond nous permet de prédire les paramètres du DRC utilisés à partir d'une observation du signal compressé. 2) les paramètres estimés sont ensuite utilisés pour réaliser une inversion paramétrique de la compression [4].

Cet article est organisé de la manière suivante. La méthode proposée est introduite dans la Section 2. Nous présentons ensuite notre protocole expérimental et les données utilisées dans la Section 3 avant de présenter nos résultats numériques inclus dans la Section 4. Nous concluons ce travail avec la Section 5 qui propose des pistes d'améliorations futures.

2 Méthode

2.1 Formulation du problème

Soit $x \in \mathbb{R}^N$ un signal à temps discret obtenu à une fréquence d'échantillonnage F_s. Nous notons $x[n], \forall n \in \{0, 1, \cdots, N-1\}$ l'échantillon du signal à la position n. Le signal compressé est obtenu en appliquant une fonction de gain g du DRC qui varie au cours du temps et dépend du signal x et des paramètres DRC $q_\theta \in \mathbb{R}^7$:

$$y[n] = x[n] \cdot g_{x,q_\theta}[n]. \tag{1}$$

$q_\theta = \left[L, R, \tau_v^{att}, \tau_v^{rel}, \tau_g^{att}, \tau_g^{rel}, p\right]$, correspond à un vecteur de paramètres liés au DRC (cf. Table 1) avec $\theta \in \{0, 1, ..., d-1\}$, le label d'un profil DRC, d étant le nombre maximum de profils considérés. Nous traitons ici le problème d'estimation aveugle de $\hat{q}_\theta$ et de $\hat{x}$ à partir de la seule observation du signal compressé y. Nous souhaitons obtenir des estimations $\hat{q}_\theta$ et $\hat{x}$ aussi proches que possible de la vérité terrain utilisée uniquement au moment de l'évaluation.

2.2 Inversion de DRC basée sur un modèle

Nous utilisons la méthode proposée dans [4], permettant d'estimer x à partir de y à condition que q_θ soit exactement connu. Cette méthode se base sur une inversion approchée de la fonction caractéristique du compresseur, reposant sur la recherche des racines de la fonction :

$$\xi_p(v[n]) = (\gamma\kappa v[n]^{-S} + \bar{\gamma} g[n-1])^p \\ (v[n]^p - \bar{\beta} v[n-1]^p) - \beta |y[n]|^p \tag{2}$$

avec $S = 1 - \frac{1}{R}$, $l = 10^{L/20}$, $\kappa = l^S$, $\gamma = 1 - e^{-\frac{2.2}{F_s \tau_g}}$, $\beta = 1 - e^{-\frac{2.2}{F_s \tau_v}}$, $\bar{\beta} = 1 - \beta$ (resp. $\bar{\gamma}$). L'enveloppe de détection est donnée par $v[n] = \sqrt[p]{\beta |x[n]|^p + \bar{\beta} v[n-1]^p}$.

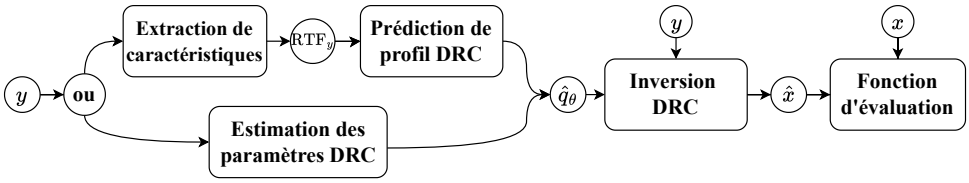

FIGURE 1 : Approche d'inversion DRC proposée.

Ainsi, l'Éq. (2) permet de retrouver les valeurs de $v[n]$. En utilisant $\tilde{x}[n] = v[n]^p$, nous pouvons déduire :

$$|x[n]| = \sqrt[p]{\tilde{x}[n] - \frac{1-\beta}{\beta}\tilde{x}[n-1]}, \tag{3}$$

ce qui permet l'estimation du gain instantané nécessaire pour inverser la DRC : $g[n]$=$\frac{|y[n]|}{|x[n]|}$ et $x[n] = \text{signe}(y[n]) \cdot |x[n]|$.

2.3 Méthode hybride proposée pour l'inversion du DRC

Nous proposons maintenant une nouvelle méthode d'inversion de DRC illustrée dans la Figure 1, qui implique une approche en deux étapes successives. En prenant le signal compressé y comme entrée, nous utilisons un modèle d'apprentissage profond supervisé qui estime les paramètres DRC q_θ de deux façons distinctes :

Prédiction du profil DRC θ : On utilise un modèle Transformeur Audio du Spectrogram (AST) [7] prenant en entrée le spectrogramme de x et qui prédit le profil θ, qui nous permet de déduire le vecteur $\hat{q}_\theta$ des paramètres correspondant au profil supposé connu. Le modèle AST a été modifié pour utiliser un Perceptron MultiCouche (PMC) à la sortie. Chaque couche cachée est suivie d'une normalisation par lots et d'une activation PReLU.

Régression des paramètres q : On utilise un modèle Encodeur d'Effets Musicaux (MEE) [8] reposant sur une architecture de type autoencodeur. Ce modèle estime directement les paramètres $\hat{q}_\theta$ à partir de y en utilisant un PMC de 4 couches en sortie.

Une fois $\hat{q}_\theta$ obtenu, la méthode d'inversion décrite dans la Section 2.2 est appliquée sur le signal y pour estimer le signal non compressé $\hat{x}$.

3 Protocole expérimental

3.1 Jeux de données

Nous utilisons 4 jeux de données : MedleyDB [9], MUSDB18-HQ [10], DAFX [11] et LibriSpeech [12]. Nous sélectionnons aléatoirement 30 morceaux de MedleyDB, d'une durée totale d'environ 1.6 heure, et sélectionné au hasard des audio de la même durée à partir d'autres jeux de données. Tous les signaux sont segmentés en clips de 5 secondes, puis nous retirons tous les segments ayant une Racine de la Moyenne des Carrés (RMS) inférieure à -30 dB. Nous avons créé 30 profils de DRC à partir des paramètres de la Table 1. Tous les profils utilisent un détecteur RMS ($p = 2$) et les autres paramètres sont choisis par un tirage aléatoire uniforme pour assurer une couverture complète de l'espace des paramètres tout en maintenant des réglages de compression réalistes. Les quatre jeux de données résultants contiennent chacun 35 867 échantillons, répartis en 31 classes (30 profils DRC et un profil neutre sans compression).

TABLE 1 : Valeurs possibles des paramètres DRC utilisés dans les 30 profils DRC considérés selon [1].

Paramètre	Description	Borne inférieure	Borne supérieure
L (dBFS)	Seuil	-60	-20
R (dB_{in} :dB_{out})	Ratio	2	15
τ_v^{att} (ms)	Attaque d'enveloppe	5	130
τ_v^{rel} (ms)	Relâchement d'enveloppe		
τ_g^{att} (ms)	Attaque de gain	10	500
τ_g^{rel} (ms)	Relâchement de gain	25	2 000
p (1 ou 2)	Type de détecteur	2	2

3.2 Paramètres expérimentaux

Nous utilisons un protocole d'apprentissage supervisé avec un ratio entraînement/test de 4 :1, garantissant l'absence de chevauchement entre les ensembles. Nous employons l'optimiseur ADAM avec un taux d'apprentissage initial de 10^{-4}, une taille des mini-batchs de 12, et un maximum de 500 epochs, avec une décroissance exponentielle du taux d'apprentissage d'un facteur $\alpha = 0.98$ par epoch. L'entraînement s'arrête après 30 epochs sans amélioration.

Pour la prédiction des profils DRC (classification), le modèle AST minimise l'entropie croisée entre les étiquettes estimées $\hat{\theta}$ et réelles θ. Pour l'estimation des paramètres DRC (régression), nous minimisons l'Erreur Quadratique Moyenne (EQM) entre les paramètres estimés $\hat{q}$ et réels q.

4 Résultats numériques

Nos méthodes sont implémentées en python utilisant la bibliothèque PyTorch[1]. Les calculs sont réalisés avec un CPU Intel Xeon W-2223 @ 3.60GHz disposant de 32 Go de RAM et un GPU Nvidia RTX4080 Super avec 16 Go de VRAM.

4.1 Résultats de classification des profils DRC

Dans la Table 2, nous comparons plusieurs choix de Représentation Temps-Fréquence (RTF) utilisées en entrée du modèle : Coefficients Cepstraux en échelle Mel (MFCC), le spectrogramme obtenu par Transformée de Fourier à court-terme (TFCT), le spectrogramme en échelle Mel (MelS) et le spectrogramme de transformée en Q constant (CQT). Selon [13], nous explorons différentes tailles d'entrée pour identifier le choix optimal. D'après la Table 2, nous observons que la TFCT de taille $(64, 431)$ fournit dans tous les cas le meilleur taux de classification correcte et un temps d'exécution plus rapide (environ 2min/epoch contre 42min pour la CQT).

Après avoir fixé la RTF de l'AST, nous évaluons l'efficacité de la profondeur du PMC utilisé pour les dernières couches de sortie sur la classification DRC. D'après la Figure 3, pour MedleyDB et MUSDB18-HQ, l'utilisation de 5 couches de PMC fournit les meilleurs résultats, tandis que pour DAFX et LibriSpeech, le meilleur choix est de 3 couches. En effet,

[1]Codes pythons disponibles sur `https://github.com/SunHaoRanCN/Inversion-de-la-Compression-Dynamique`

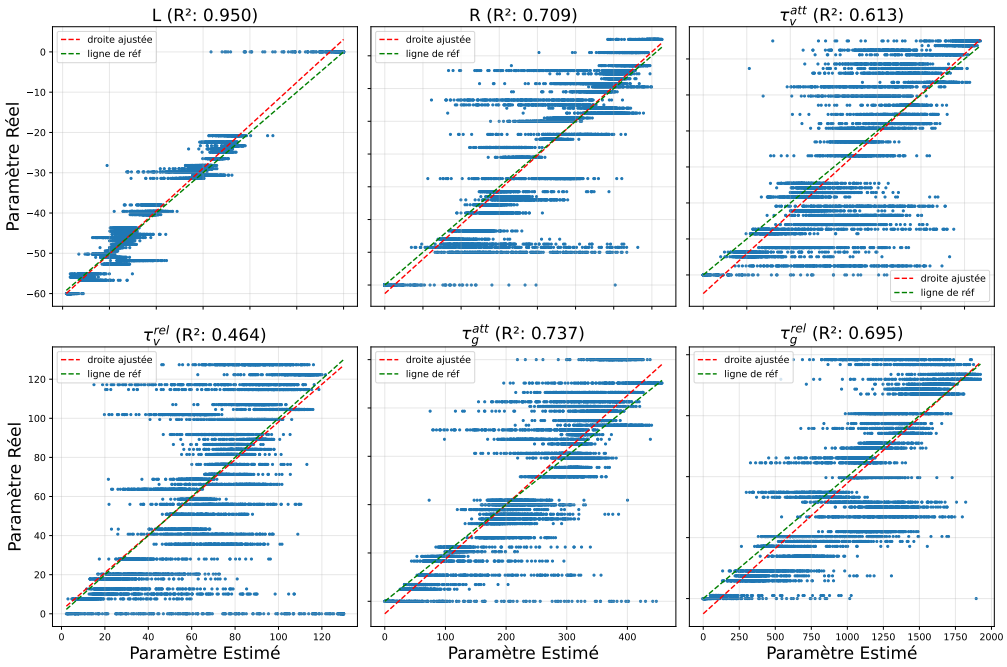

FIGURE 2 : Nuage de points et coefficient de détermination R^2 pour l'estimation de chaque paramètre DRC dans Table 1 (sauf p) par la méthode MEE appliquée sur le jeux de données MedleyDB. En rouge la droite de régression linéaire estimée et en vert la droite de référence ($y = x$).

TABLE 2 : Comparaison des performances de classification (précision) reposant sur le modèle AST (sans PMC supplémentaire) en fonction de l'entrée pour chaque jeu de données, nous affichons ici uniquement les tailles d'entrée qui offrent les meilleures performances.

Jeu de données	Entrée	Taille	Temps/epoch(h)	Pré.
MedleyDB	MFCC	(128, 431)	0,093	0,66
	STFT	(64, 431)	**0,037**	**0,82**
	MelS	(128, 431)	0,074	0,81
	CQT	(128, 862)	0,790	0,55
MUSDB18-HQ	MFCC	(128, 431)	0,093	0,65
	STFT	(64, 431)	**0,037**	**0,80**
	MelS	(128, 431)	0,074	0,78
	CQT	(128, 862)	0,790	0,53
DAFX	MFCC	(64, 431)	0,091	0,72
	STFT	(64, 431)	**0,036**	**0,84**
	MelS	(64, 431)	0,072	0,83
	CQT	(64, 862)	0,750	0,66
LibriSpeech	MFCC	(64, 431)	0,089	0,73
	STFT	(64, 431)	**0,035**	**0,85**
	MelS	(64, 431)	0,070	0,84
	CQT	(64, 862)	0,710	0,67

TABLE 3 : Comparaison de l'estimation des paramètres DRC et du temps de calcul (en heures), pour les modèles MEE et TFE.

(a) MEE

Jeu de données	Temps/epoch(h)	EQM
MedleyDB	0,084	**0.031**
MUSDB18-HQ	0,087	**0.039**
DAFX	0,065	**0.025**
Librispeech	0,040	**0.012**

(b) TFE

Jeu de données	Temps/epoch(h)	EQM
MedleyDB	**0,032**	0,058
Musdb18-hq	**0,038**	0,058
DAFX	**0,022**	0,054
Librispeech	**0,015**	0,052

la fréquence d'échantillonnage des deux premiers jeux de données est plus élevée.

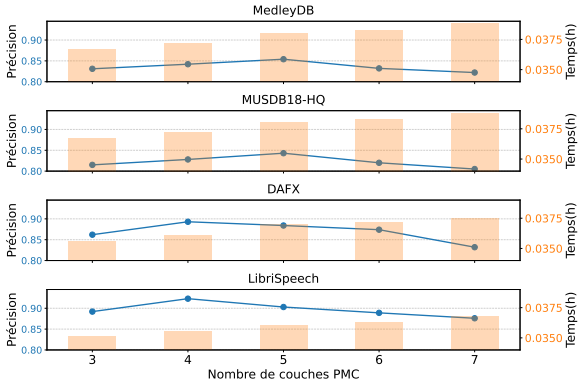

FIGURE 3 : Impact du nombre de couches PMC sur la classification des profils DRC utilisant le modèle AST, où le graphique à barres jaunes représente le temps d'apprentissage moyen pour une époque et la ligne bleue représente la précision obtenue.

4.2 Résultats d'estimation des paramètres DRC

Dans cette expérience qui considère une tâche de régression, on utilise le modèle MEE [8] qui traite directement le signal temporel en entrée. Comme le type de détecteur de tous les profils DRC dans ce travail est fixé à p=2, nous nous concentrons sur l'estimation des 6 autres paramètres.

Nous comparons cette méthode avec un second modèle Encodeur Temps-Fréquence (TFE) [14] qui utilise en entrée le spectrogramme du signal de taille $(64, 431)$. Dans cette tâche, nous minimisons l'EQM des paramètres estimés $\hat{q}_\theta$ et des paramètres réels q_θ.

La Table 3 montre que le modèle MEE obtient une EQM plus faible et fournit de meilleurs résultats que TFE pour tous les jeux de données. En revanche, le modèle TFE est en moyenne 2.6 fois plus rapide que le modèle MEE.

La Figure 2 présente les résultats distincts pour chaque paramètre DRC en utilisant la méthode MEE sur MedleyDB. On

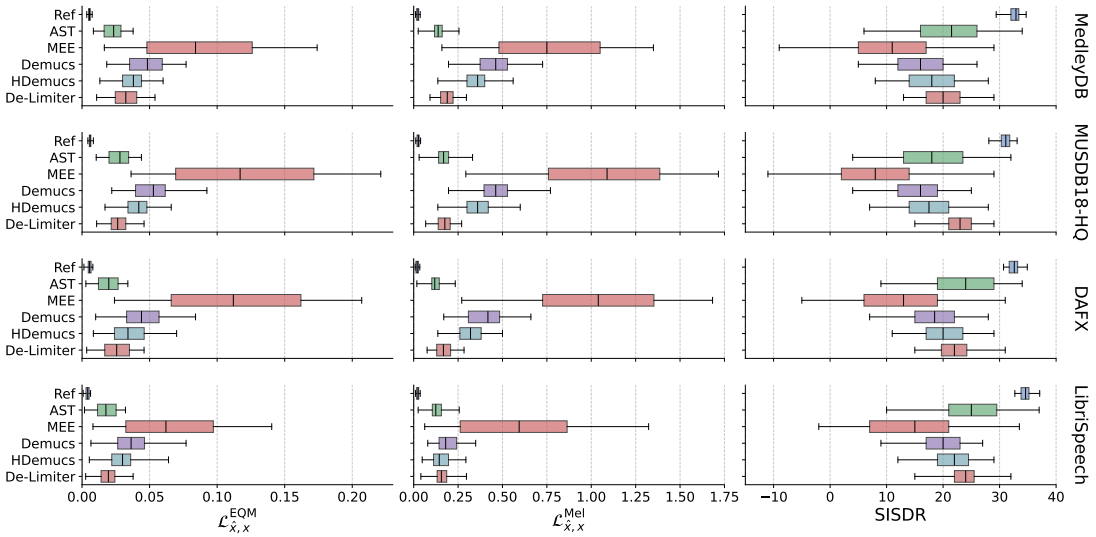

FIGURE 4 : Comparaison des performances de la tâche d'inversion DRC. Les modèles proposés (AST, MEE) sont comparés à la référence (Ref) et les modèles de l'état de l'art (Demucs [15], HDemucs [16], De-limiter [5]).

remarque que le seuil L est très bien estimé avec un coefficient $R^2 = 0,95$. Le ratio R et les paramètres de lissage d'enveloppe τ_g sont assez bien identifiés avec $R^2 \approx 0,7$. Cependant, les paramètres de lissage du seuil de détection τ_v sont plus difficiles à estimer par la méthode.

4.3 Résultats d'inversion DRC

Enfin, nous comparons la méthode complète proposée avec plusieurs techniques de l'état de l'art reposant sur les réseaux de neurones profonds : Demucs [15] et Hybrid Demucs [16] ont été utilisés à l'origine pour la séparation des sources, cependant, ils ont aussi montré de bonne performances dans les tâches de restauration audio [17], ainsi que De-limiter [5].

Nous comparons la qualité de reconstruction du signal exprimée en termes d'EQM et de distance Mel :

$$\mathcal{L}^{\text{Mel}}_{\hat{x},x} = \| \ln(|\text{Mel}_{\hat{x}}|) - \ln(|\text{Mel}_x|)\|_2, \tag{4}$$

obtenue par la méthode présentée dans la Section 2.2 en utilisant les paramètres $\hat{q}$ fournis respectivement par les méthodes AST et MEE. De plus, nous utilisons les paramètres de vérité terrain q_θ pour effectuer l'inversion DRC comme référence. Dans cette expérience, toutes les méthodes de l'état de l'art utilisent les implémentations fournies par les auteurs originaux. Demucs et HDemucs sont réentraînés sur nos jeux de données et De-limiter, utilise directement le modèle préentraîné fourni par les auteurs.

Figure 4 montre que le modèle AST fournit de meilleurs résultats sur la plupart des jeux de données, tandis que De-limiter atteint un meilleur équilibre sur MUSDB18-HQ. Bien que la performance du modèle MEE semble insuffisante, les résultats obtenus indiquent une faisabilité de cette approche à condition d'améliorer la précision d'estimation des paramètres τ_v qui font défaut.

Le modèle AST est plus performant que le modèle MEE car : premièrement, la tâche de classification est naturellement plus simple que la tâche de régression ; deuxièmement, une erreur d'estimation est inévitable dans la tâche de régression ; et troisièmement, des paramètres différents peuvent parfois produire le même effet de compression, ce qui complexifie encore la tâche de régression.

5 Conclusion

Nous avons présenté une nouvelle méthode qui combine une approche modèle avec de l'apprentissage profond pour inverser un effet audio de compression dynamique. Nos résultats expérimentaux sont prometteurs et semblent compétitifs avec l'état de l'art. De plus, notre approche permet d'estimer les profils et les paramètres DRC directement à partir du seul signal observé. Nos efforts se porteront ensuite sur l'amélioration des performances du modèle MEE durant la tâche de régression des paramètres DRC.

Références

[1] U. Zölzer, X. Amatriain, D. Arfib, J. Bonada, G. De Poli, P. Dutilleux, G. Evangelista, F. Keiler, A. Loscos, D. Rocchesso, *et al.*, *DAFX-Digital audio effects*. John Wiley & Sons, 2002.

[2] D. Barchiesi and J. Reiss, "Reverse engineering of a mix," *Journal of the Audio Engineering Society*, pp. 563–576, 2010.

[3] B. Lachaise and L. Daudet, "Inverting dynamics compression with minimal side information," in *Proc. DAFx*, 2008.

[4] S. Gorlow and J. D. Reiss, "Model-based inversion of dynamic range compression," *IEEE Trans. on Audio, Speech, and Language Processing*, pp. 1434–1444, 2013.

[5] C.-B. Jeon and K. Lee, "Music de-limiter networks via sample-wise gain inversion," in *Proc. IEEE WASPAA*, pp. 1–5, 2023.

[6] M. Rice, C. J. Steinmetz, G. Fazekas, and J. D. Reiss, "General purpose audio effect removal," in *Proc. IEEE WASPAA*, pp. 1–5, 2023.

[7] Y. Gong, Y.-A. Chung, and J. Glass, "AST : Audio Spectrogram Transformer," in *Proc. Interspeech*, pp. 571–575, 2021.

[8] J. Koo, S. Paik, and K. Lee, "End-to-end music remastering system using self-supervised and adversarial training," in *Proc. IEEE ICASSP*, pp. 4608–4612, 2022.

[9] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb : A multitrack dataset for annotation-intensive mir research.," in *Proc. ISMIR*, pp. 155–160, 2014.

[10] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-hq - an uncompressed version of musdb18," 2019.

[11] D. Tardieu, E. Detruty, and G. Peeters, "Production effect : Audio features for recordings techniques description and decade prediction," in *Proc. DAFx*, pp. 441–446, 2011.

[12] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech : an ASR corpus based on public domain audio books," in *Proc. IEEE ICASSP*, pp. 5206–5210, 2015.

[13] M. Huzaifah, "Comparison of time-frequency representations for environmental sound classification using convolutional neural networks," *arXiv :1706.07156*, 2017.

[14] C. Peladeau and G. Peeters, "Blind estimation of audio effects using an auto-encoder approach and differentiable digital signal processing," in *Proc. IEEE ICASSP*, pp. 856–860, 2024.

[15] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv :1911.13254*, 2019.

[16] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *Proc. IEEE ICASSP*, 2023.

[17] J. Imort, G. Fabbro, M. A. M. Ramírez, S. Uhlich, Y. Koyama, and Y. Mitsufuji, "Removing distortion effects in music using deep neural networks," *arXiv :2202.01664*, 2022.