

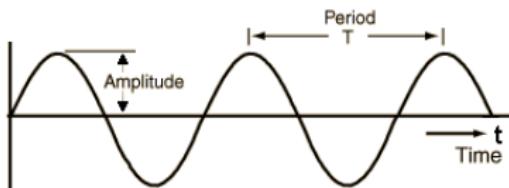
# AIT AUDIO INPAINTING WITH IMPLICIT NEURAL REPRESENTATION

Saturday 26<sup>th</sup> March, 2022

Youva ADDAD, Kamel NAIT SLIMANI

# Introduction

# Qu'est-ce que l'audio?



Signal répétitif simple montrant l'amplitude en fonction du temps

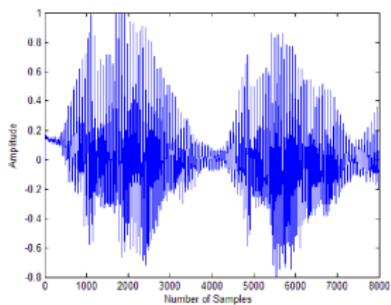


$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ \dots \ x_t \ \dots]^T$$

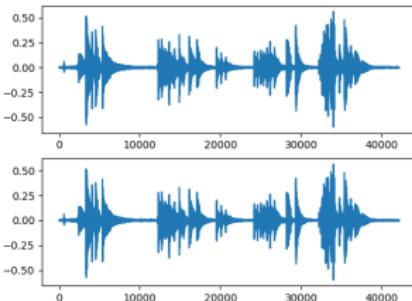
ou  $x$  est la séquence composant l'audio et  $x_t$  est la hauteur de l'onde au temps  $t$

# Qu'est-ce que l'audio?

Séquences de valeurs digitales échantillonnées à partir d'un signal analogique qui est représenté par une amplitude.



One Channel audio (mono)



Multi channel audio (stereo)

Comment peut-on traiter ces données ?

## Spectrogrammes:

### Signal

Puisqu'un signal produit des sons différents lorsqu'il varie dans le temps, ses fréquences constitutives varient également dans le temps. En d'autres termes, son spectre varie avec le temps.

### Spectrogramme

Un spectrogramme d'un signal trace son spectre dans le temps et ressemble à une "photographie" du signal. Il trace le temps sur l'axe des x et la fréquence sur l'axe des y. C'est comme si nous prenions le Spectre encore et encore à différents moments dans le temps, puis que nous les réunissions tous ensemble en une seule intrigue. Il utilise différentes couleurs pour indiquer l'amplitude ou la force de chaque fréquence

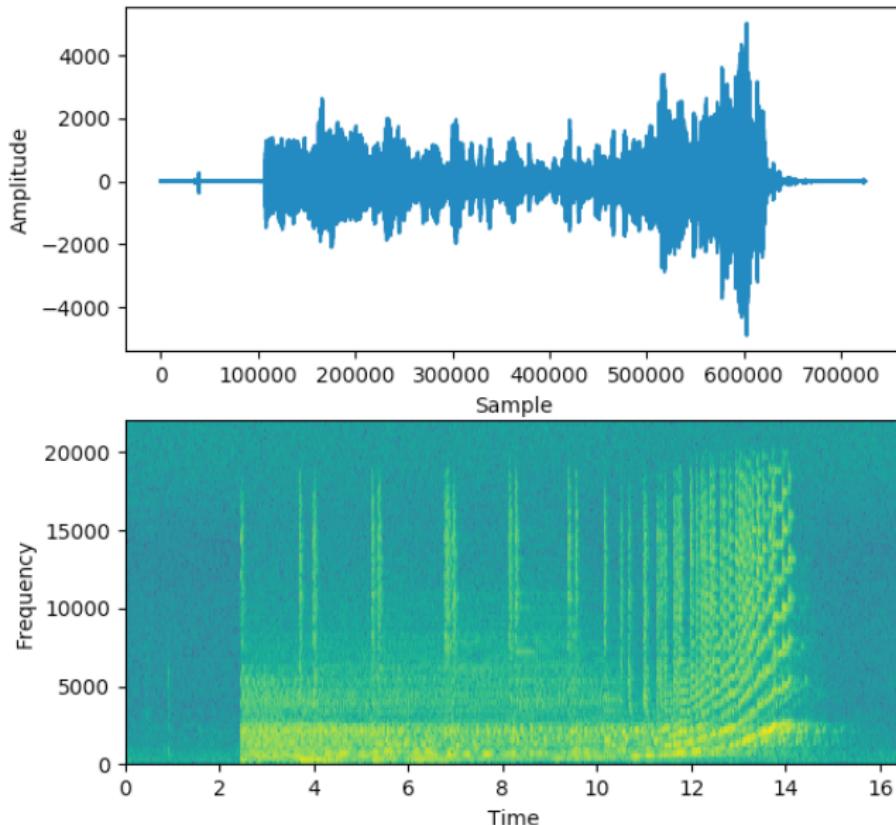
# Short-time Fourier transform (STFT)

## STFT

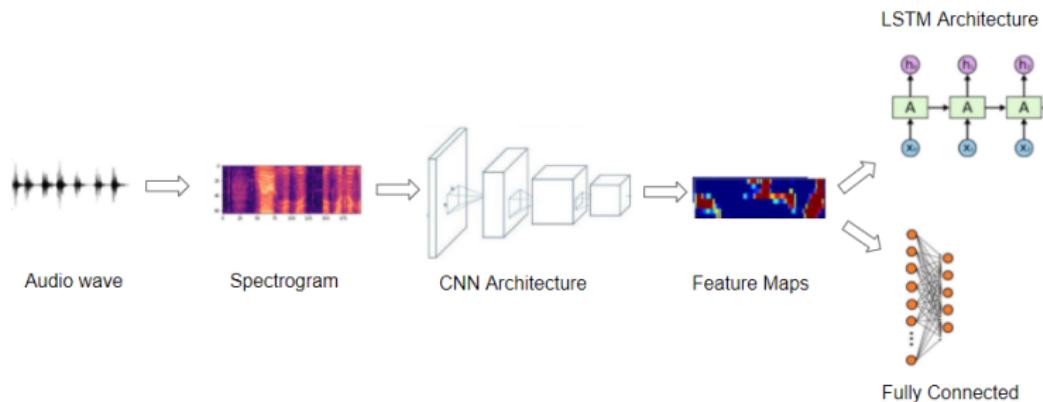
Est une transformation liée aux transformées de Fourier utilisée pour déterminer la fréquence sinusoïdale et la phase d'une section locale d'un signal. Le carré de son module donne le spectrogramme,  $\text{spectrogram}(t, \omega) = |\text{STFT}(t, \omega)|^2$

$$\text{STFT}\{x(t)\}(\tau, \omega) \equiv X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-i\omega t} dt$$

# Spectrogramme



# Modèles deep learning pour l'audio



## Note:

Lors de la modélisation de l'audio en tant que représentation temps-fréquence (Spectrogramme), la résolution temporelle est un paramètre du modèle.

## Difficultés des données audio

- Données séquentielles fortement dépendantes,  
 $p(\mathbf{x}) = \prod_{t=1}^T p(x_t | x_1, \dots, x_{t-1})$
- Suite d'éléments continues difficile d'en extraire de l'information
- Très longues séquences de données pour une représentation significative de quelques secondes d'audio
- Corrélations à la fois aux petites échelles de temps et aux grandes échelles de temps.
- 16 bits d'audio signifie 65 536 probabilités par pas de temps.
- Tonnes d'échantillons: 16 kHz = 16000 échantillons par seconde

# Audio Inpainting

# Audio Inpainting

Les Signaux audio souffrent des pertes de données que ça soit lors de l'enregistrement, perte pendant la transmission ou bien des parties de signaux analogiques perdues, la tâche de restauration de ces pertes s'appelle l'audio inpainting, cette notion est introduite par **Adler et al.** en 2012

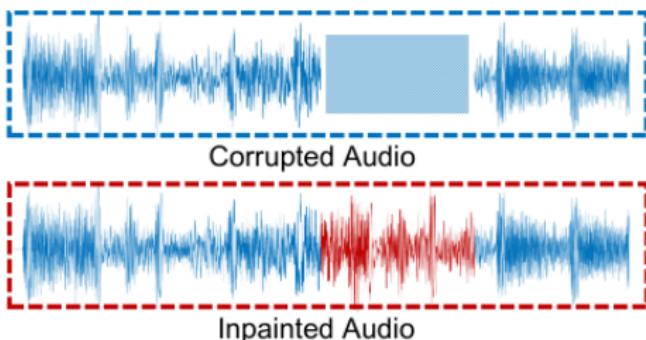


Figure 1: Inpainted audio sequence

# Audio Inpainting

## Related Work:

Le terme "audio inpainting" a été introduit par **Adler et al.** 2012 pour décrire une large classe de problèmes inverses en traitement audio. Leur propre travail, cependant, a principalement étudié la restauration des gaps dans les signaux audio. Généralement, les problèmes d'inpainting audio concernent l'audio représenté sous forme de données dans un certain domaine de caractéristiques et supposent que des morceaux de ces données sont corrompus, ce qui produit des gaps dans la représentation.

L'inpainting audio est un défi important en raison de la propriété de l'audio d'un taux d'échantillonnage élevé et d'une dépendance à longue portée.

# Formalisation du problème

## Formalisation

Soit  $x$  le vecteur de la séquence de l'audio qui est de taille  $T$ , donc  $\mathbf{x} = [x_1 \ x_2 \ x_3 \ \dots \ x_t \ \dots \ x_T]^T$ , supposons sans perte de généralité que les composantes du vecteur de l'audio  $x_t$  jusqu'à  $x_{t+k}$  est le vecteur gaps.

Toutes les composantes étant dépendantes entre elles nous aurons donc à maximiser la vraisemblance suivante:

$$\begin{aligned} P(x_1, \dots, x_T) &= \\ P(x_1, \dots, x_{t-1}, x_{t+k+1}, \dots, x_T)P(x_t, \dots, x_{t+k} | x_{t' \neq t \dots t+k}) \end{aligned}$$

## Formalisation du problème

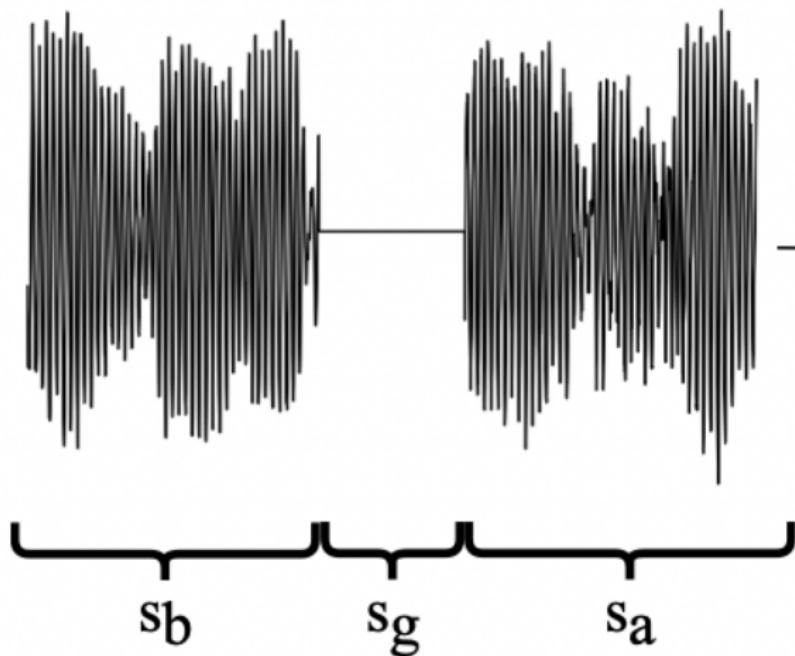


Figure 2:  $s_a$  et  $s_b$  représente le contexte et  $s_g$  représente le gap

# État de l'art

# A context encoder for audio inpainting

## Main Idea

- En considérant le signal audio  $s$  composé du gap  $s_g$  et des signaux de contexte avant et après l'espace,  $s_b$  et  $s_a$  respectivement
- le modèle est composé est un pipeline codeur-décodeur alimenté avec les coefficients TF des informations de contexte,  $s_b$  et  $s_a$ , le réseau est composé uniquement de couches convolutives, des FCL et des unités linéaires rectifiées (ReLU).
- Les coefficients TF sont obtenus à partir d'une représentation inversible, à savoir une transformée de Fourier courte durée redondante (STFT).

# A context encoder for audio inpainting

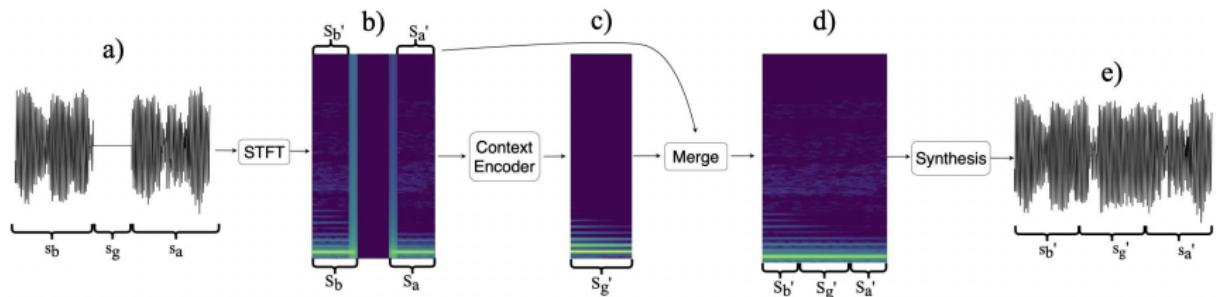


Figure 3: The end-to-end architecture

# A context encoder for audio inpainting

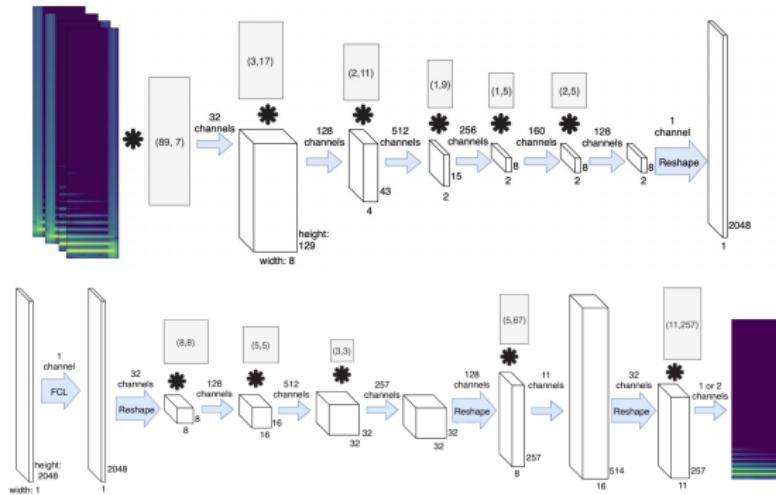


Figure 4: L'encodeur est un réseau convolutif à six couches suivi d'un reshaping, L'architecture du décodeur pour le réseau complexe et d'amplitude produisant respectivement un et deux canaux de coefficients TF

# Audio inpainting with generative adversarial network

## Main Idea

- Utilise Wasserstein Generative Adversarial Network (WGAN) pour générer du contenu audio manquant qui, dans son contexte, (statistiquement similaire) au son et aux frontières voisines.
- Les GAN sont capables de produire des signaux statistiquement similaires (de l'ordre 500 - 550 ms) sans qu'il soit nécessaire de minimiser une perte.
- L'approche appliquée ici est l'extraction des frontières courte portée et des frontières longue portée du contenu audio manquant, En combinant les deux frontières qui se chevauchent, nous pouvons avoir davantage d'informations corrélées aux données manquantes et les utiliser pour former notre modèle.

[https://github.com/nperraud/gan\\\_audio\\\_inpainting](https://github.com/nperraud/gan\_audio\_inpainting)

# Audio inpainting with generative adversarial network

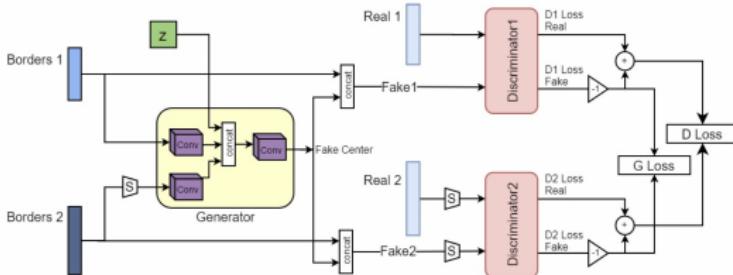


Figure 5: Dual Discriminator Wasserstein GAN (D2WGSN).

## Wasserstein loss

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|].$$

Intuitivement,  $\gamma(x, y)$  indique combien de « masse » doit être transportée de  $x$  vers  $y$  pour transformer les distributions  $P_r$  en distribution  $P_g$ .

## GACELA "A generative adversarial context encoder for long audio inpainting"

### Main Idea

- Un réseau antagoniste génératif (GAN) conçu pour restaurer les données audio musicales manquantes d'une durée comprise entre des centaines de millisecondes et quelques secondes, c'est-à-dire pour effectuer un Inpainting audio à long intervalle (allant jusqu'à 1.5 seconde).
- Le GAN est capable de modéliser la distribution des remplacements des gaps possibles au lieu de produire un seul candidat.

<https://github.com/andimarafioti/GACELA>

# GACELA : Modèle de génération

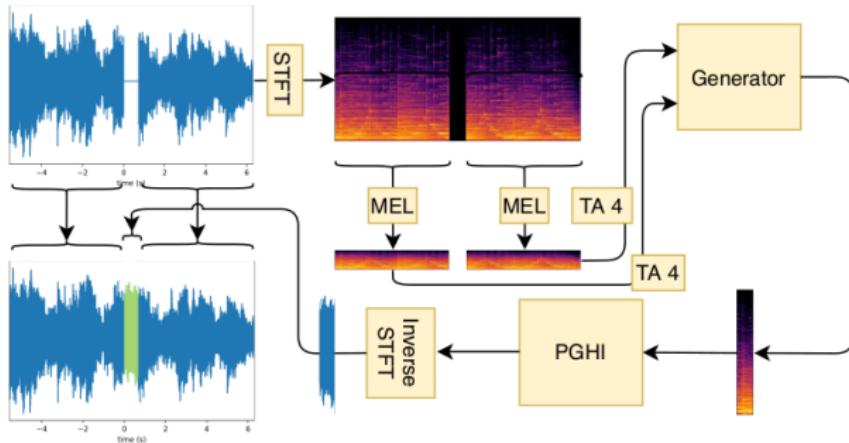
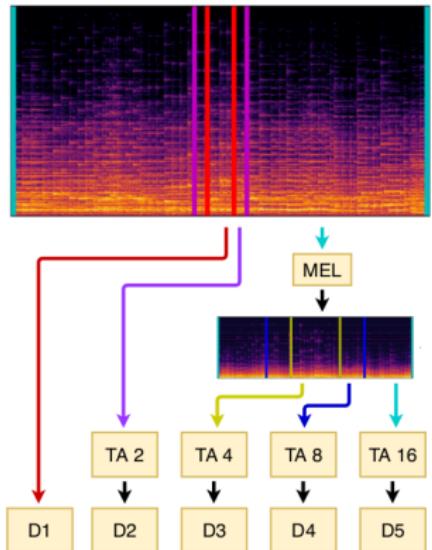


Figure 6: GACELA end-to-end audio generation system, PGHI Constructs a candidate phase for a given log- magnitude STFT, MEL Computes a mel-scale spectrogram, Time-Averaging (TA X): Reduces the time dimension of a log-magnitude STFT

# GACELA : Modèle d'apprentissage



Archit  cture du discriminateur. D1-D5 represen-  
tent des descriminateurs individuelles, leur recep-  
tive fields sont anot  s en couleur

# GACELA : Solutions et Limites

## Solutions

- Les différents discriminateurs en parallèle avec des Réceptifs Fields plus importants permettent de capturer de l'information des plus grands intervalles de temps.
- Intègre de la multimodalité, la reconstruction n'est pas seulement conditionnée sur les frontières du gap, mais aussi par la variable latente du gan conditionnel ce qui permet à l'utilisateur de plus personnaliser la génération.

## Limites

- 1.5 seconde de restitution reste toujours peu par rapport au cas réels au quels nous pouvons être confrontés.
- Baisse de performance sur d'autres styles de musique plus compliqués.

# Le Modèle Proposé

# Fourier Convolution Layer

## Audio Features

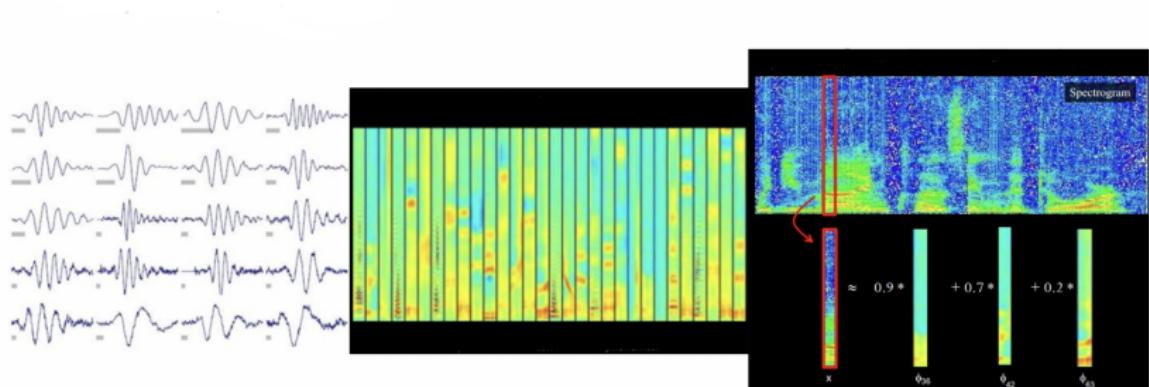
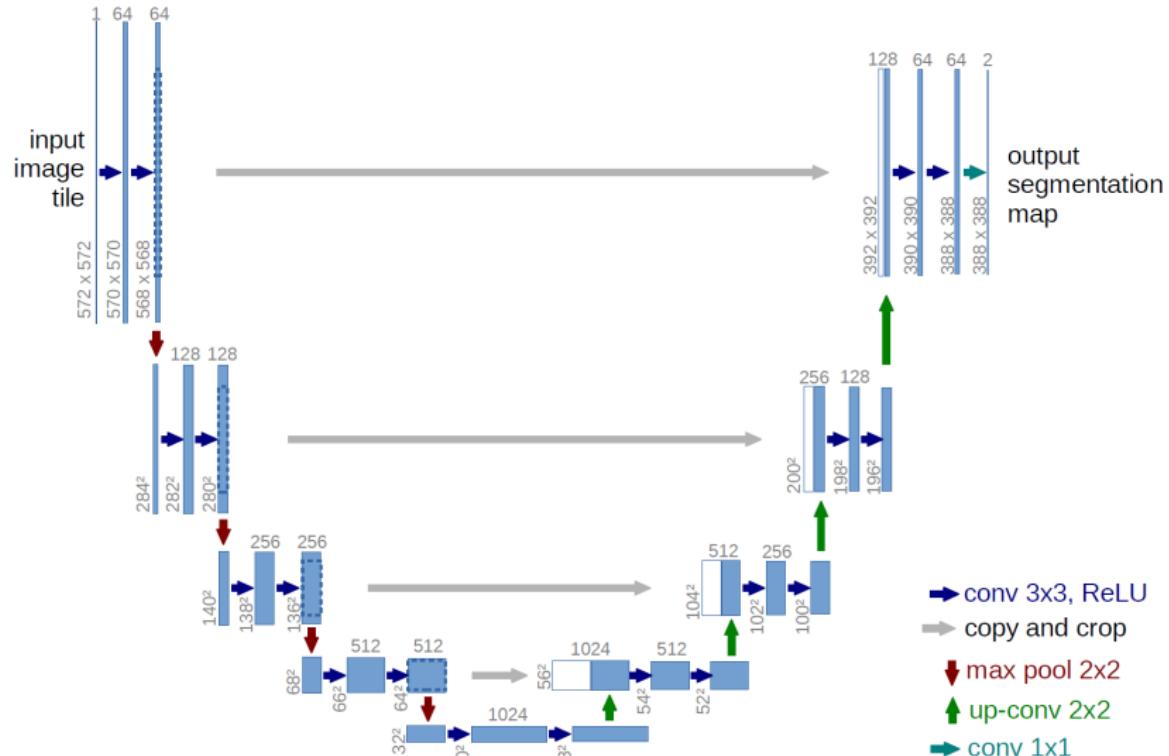


Figure 7: Audio features

## U-Net



## Code

```
class U_Audio(nn.Module):
    def __init__(self, in_channels, out_channels):
        super(U_Audio, self).__init__()
        self.in_channels = in_channels
        self.out_channels = out_channels

        self.down0 = Down(in_channels, 64) # Conv1d
        self.down1 = Down(64, 128)
        self.down2 = Down(128, 256)
        self.down3 = Down(256, 512)
        self.multihead_attn = MultiHeadAttention(512)
        self.up1 = Up(512, 256) #MultiHead + Conv
        self.up2 = Up(256, 128) #MultiHead + Conv
        self.up3 = Up(128, 64) #MultiHead + Conv
        self.proj = Conv1d(64, self.out_channels) #Convolution directement a Mel

    def forward(self, x):
        x1 = self.down0(x)
        x2 = self.down1(x1)
        x3 = self.down2(x2)
        x4 = self.down3(x3)
        x4 = self.multihead_attn(x4)
        x = self.up1(x4, x3)
        x = self.up2(x, x2)
        x = self.up3(x, x1)
        output = self.proj(x)
        return output
```

Figure 9: Code

# Reconstruction

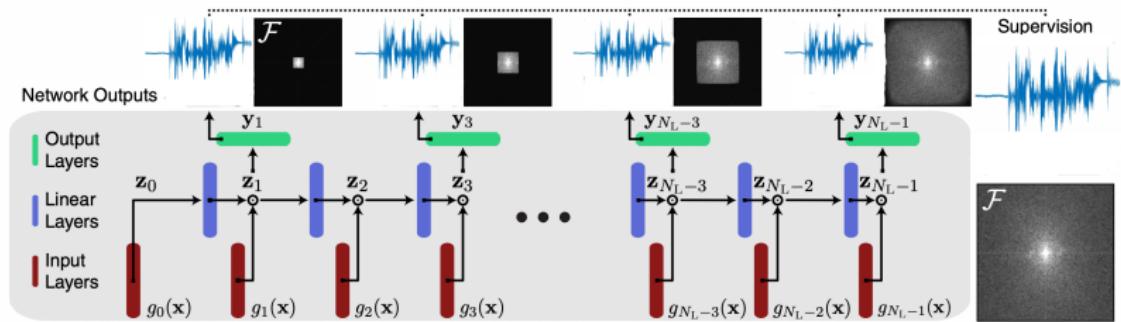


Figure 10: Reconstruction

## Détaille

## Précision:

- $z_0 = g_0(x)$
- $z_i = g_i(x) \circ (W_i z_{i-1} + b_i), \quad 0 \leq i < N_L.$
- $y_i = W_i^{\text{out}} z_i + b_i^{\text{out}}$ .
- $g_i(x) = \sin(\omega_i x + \phi_i).$

- Modèle entraîné en superviser end-to-end.
- Le module Multi-Head Self-Attention à la fin de l'encoder du U-Net donne accès aux receptive fields contenant tout le spectrogramme.
- Les autres MSA sont dédiés à combiner la richesse sémantique des cartes d'entités de haut niveau avec la haute résolution.

Surmonter l'effet néfaste des non-linéarités traditionnelles comme ReLU/tanh sur la modélisation des détails fins et la dérivée d'ordre supérieur des signaux d'entrée.

# Evaluation

# Données

Pour l'évaluation, nous avons testé notre modèle sur des différents datasets de différents styles de musique et de différents niveaux de complexité:

- Simple Midi musique écrite à la main Lakh MIDI
- Séquences Midi extraites à partir de "International Piano Competition" Maestro dataset
- Grand Piano réel Enregistré.
- Musique mixte de Différents genres

# Methods d'évaluation

Pour évaluer le modèle et le comparer à l'état de l'art, nous avons utilisé deux méthodes

- L'erreur de reconstruction la distance entre le spectrogramme de la séquence générée et la séquence original.
- ODG Une méthode basée sur la perceptibilité de la différence entre deux séquences audio selon le tableau suivant.

ODG	Impairment
0	Imperceptible
-1	Perceptible, but not annoying
-2	Slightly annoying
-3	Annoying
-4	Very annoying

Table 1: ODGs Score Interpretation

# Evaluations

	CEAI		AIGAN		GACELA		AIT	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
Simple MIDI	-1.91	0.21	-1.73	0.35	-0.109	0.005	<b>-0.085</b>	0.005
Rec MIDI	-2.21	0.31	-2.25	0.23	-0.125	0.008	<b>-0.094</b>	0.007
Rec PIANO	-2.9	0.19	-2.7	0.3	-0.231	0.012	<b>-0.115</b>	0.014
wold music	-3.53	0.3	-3.42	0.41	-0.613	0.051	<b>-0.371</b>	0.045

Table 1: Moyenne et variance des Scores ODG sur les différents jeux de données pour 500ms de reconstruction

	Reconstruction Loss			
	CEAI	AIGAN	GACELA	AIT
Simple MIDI	4.32	4.21	2.24	<b>1.91</b>
Rec MIDI	5.12	5.33	2.34	<b>2.03</b>
Rec PIANO	7.23	6.99	2.67	<b>2.19</b>
wold music	10.42	10.17	3.65	<b>2.93</b>

Table 2: L'erreur de construction de spectrogramme sur les différents jeux de données sur 500 ms

	ODG Mean Score			
	CEAI	AIGAN	GACELA	AIT
50 ms	-0.091	-0.101	<b>-0.012</b>	-0.021
200 ms	-1.561	-1.443	-0.134	<b>-0.098</b>
500 ms	-2.912	-2.732	-0.231	<b>-0.115</b>
1 s	-3.821	-3.850	-1.131	<b>-0.315</b>
5 s	-4	-4	-3.012	<b>-1.215</b>

Table 3: Moyennes du score ODG sur les différentes tailles de gap sur les Données PIANO

# Conclusion

# References

-  Adler, Amir, et al. « Audio Inpainting ». IEEE Transactions on Audio, Speech and Language Processing, vol. 20, no 3, mars 2012, p. 922. hal.inria.fr,  
<https://doi.org/10.1109/TASL.2011.2168211>.
-  Ebner, P. P., et A. Eltelt. « Audio inpainting with generative adversarial network ». arXiv:2003.07704 [cs, eess, stat], mars 2020. arXiv.org, <http://arxiv.org/abs/2003.07704>.
-  Marafioti, Andrés, et al. « A context encoder for audio inpainting ». IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no 12, décembre 2019, p. 2362-72. arXiv.org,  
<https://doi.org/10.1109/TASLP.2019.2947232>.
-  Donahue, Chris, et al. « Adversarial Audio Synthesis ». arXiv:1802.04208 [cs], février 2019. arXiv.org,  
<http://arxiv.org/abs/1802.04208>.

- 
- Sitzmann, Vincent, et al. « Implicit Neural Representations with Periodic Activation Functions ». arXiv:2006.09661 [cs, eess], juin 2020. arXiv.org,
- <http://arxiv.org/abs/2006.09661>
- .