

Master M2 - DataScience

Audio and music information retrieval

Lecture on
Audio/Music synthesis

Gaël RICHARD

Télécom Paris

March 2023

« Licence de droits d'usage" http://formation.enst.fr/licences/pedago_sans.html



Content

■ Introduction

- A brief historical overview

■ The main approaches for audio synthesis:

- Analysis/synthesis methods (signal models, sound production models,...)
- Physical models
- Data-driven models

■ A focus on (deep) neural audio synthesis

- Auto-regressive models
- Variational AutoEncoders
- Generative adversarial models
- Normalizing flows - Diffusion models
- Towards hybrid deep learning (DDSP, DDX7,...)

■ Evaluation



Lecture 8: What you need to know

■ Audio synthesis methods

- What is additive synthesis ?
- Explain the principle of FM synthesis
- Explain the main principle of VAEs. How is represented the latent space ? What is the difference with VQ-VAEs ?
- Explain the main principle of GANs. What is conditioning and how it can be used for audio synthesis?
- What is DDSP ? Why it is interesting ? What is the multiscale spectral loss ?

■ Evaluation

- How to evaluate music synthesis ?
- What are Inception-based metrics ?
- What is a subjective evaluation ? What is MOS ?



Audio synthesis

Audio synthesis = processes and algorithms which produce musical sounds

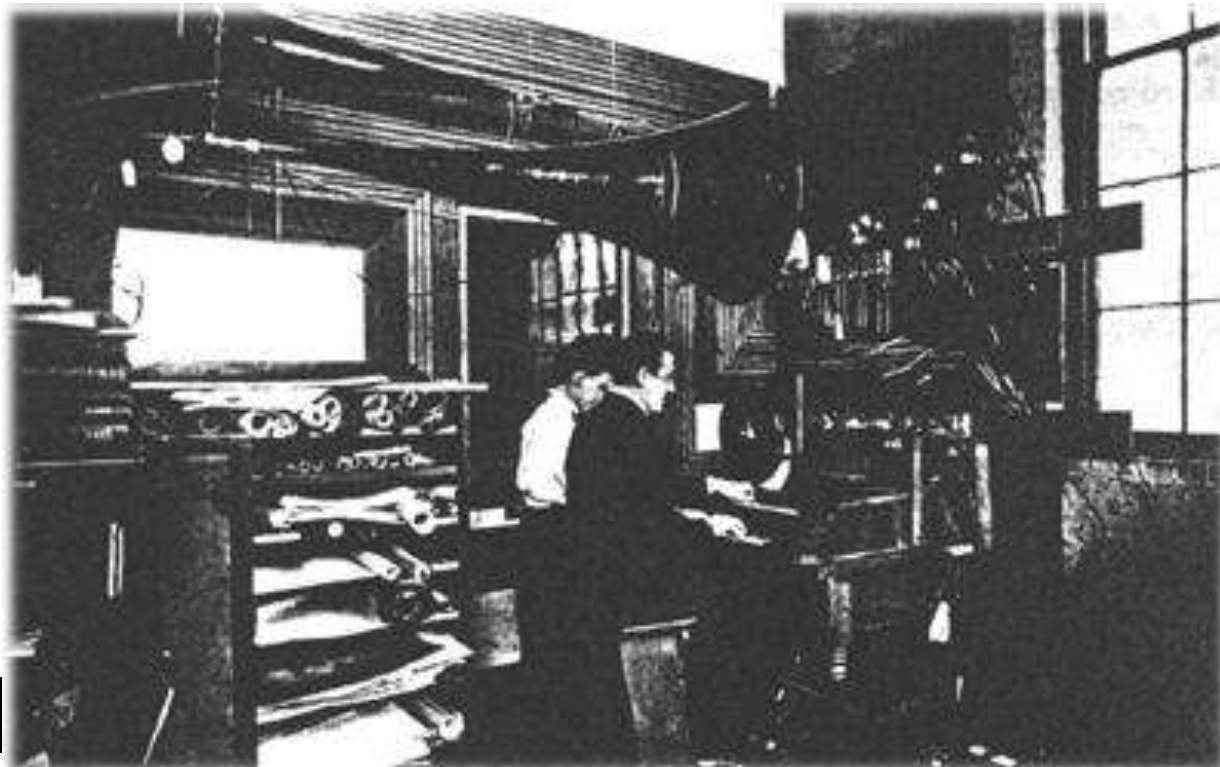
■ Interest of audio synthesis:

- Synthesizers (*« to play any instrument from a keyboard... or a computer »*)
- Support for music instrument learning
- Karaoke
- Instrument design and making (*« e.g. to hear the instrument before building it »*)
- Better understanding the physics of musical instruments



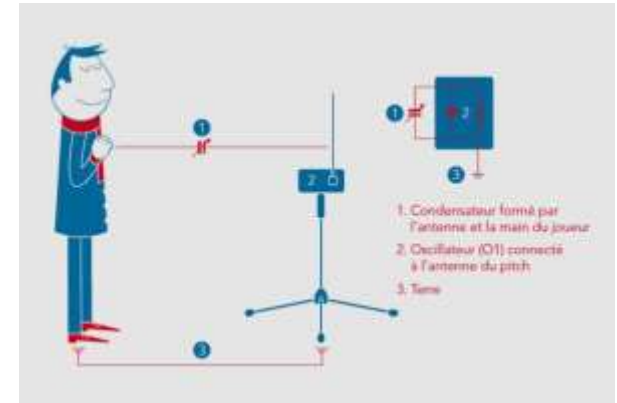
Synthesis: a « short » history

- 1897: The Telharmonium (or Dynamophone), Thaddeus Cahill
- Polyphonic instrument: can produce sounds of any frequency and intensity, with all their harmonics
- Oscillators: alternators driven by electric engines.....
- A few numbers: 200.000 \$, 200 tons, 18 meter wide,.....



Synthesis: a « short » history

■ 1920: The Theremin



■ 1928: « Ondes Martenot » (excerpt from Turangalila (6th mvnt.), Jardin du sommeil d'amour, Messian)



■ 1930: Trautonium ...



■ 1935 : Organ Hammond

■ 1954: First synthesizer: RCA Mark I (Harry F.Olsen, Columbia-Princeton)

■

■ The commercial era

- Synclavier (1972), Yamaha DX7 (1983),



https://www.easyzic.com/dossiers/un-peu-d-histoire_h393.html

<https://120years.net/wordpress/> (an extensive list of historical synthesizers)



Audio synthesis: a variety of approaches

- Analysis/synthesis of a signal model " *use of waveforms that are then used to make sound* ".
 - Additive Synthesis
 - Granular Synthesis
 - Wavetable Synthesis
- Analysis/synthesis of a production model " *using a model, source elements and operators* ".
 - FM Synthesis ("Frequency Modulation")
 - Source-filter synthesis
- Physical model " *aims at reproducing the real behavior of the instrument through a physical model* ".
 - Synthesis by discretization of propagation equations
 - Synthesis by transmission lines (ex. Karplus-Strong)
- Machine-learning based models " *uses large amount of data and machine learning to generate sounds* ".
 - Deep neural synthesis



Analysis/synthesis of a signal model

■ Additive synthesis

- Sum of elementary waveforms:
 - Sinusoids
 - Time-frequency grains
 - FOF (Formant Wave Forms)

■ Some instruments

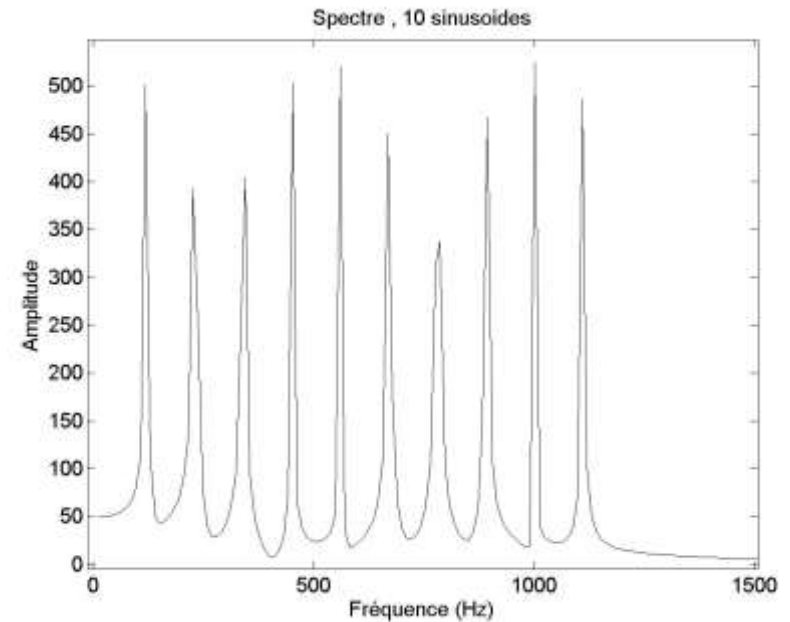
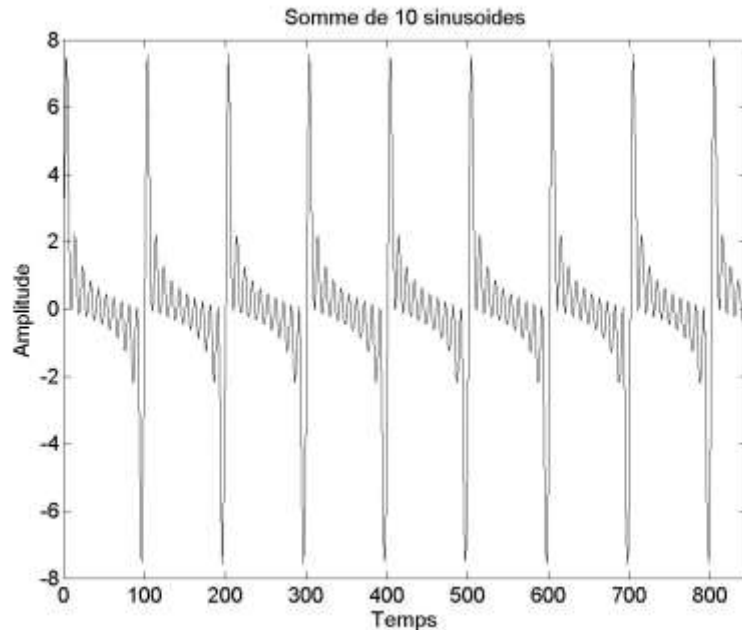
- The classic organ..
- Organ Hammond
- Synclavier



Analysis/synthesis of a signal model

■ Additive synthesis

- Synthesis performed by adding sinusoids in frequency, amplitude and phase







Analysis/synthesis of a signal model

■ An example on Piano

- Use of a decomposition « Sum of sinusoids + noise »

⇒ Example on a piano signal

- Original signal: 
- Transposed by a third: 
- Signal S (« Sum of sinusoids with vibrato effect »): 
- Signal N (Noise): 



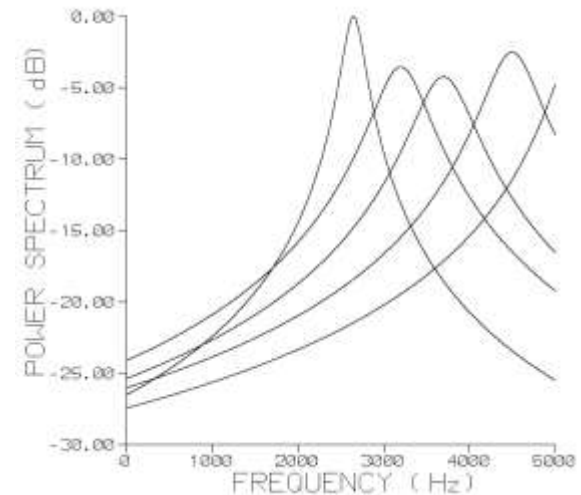
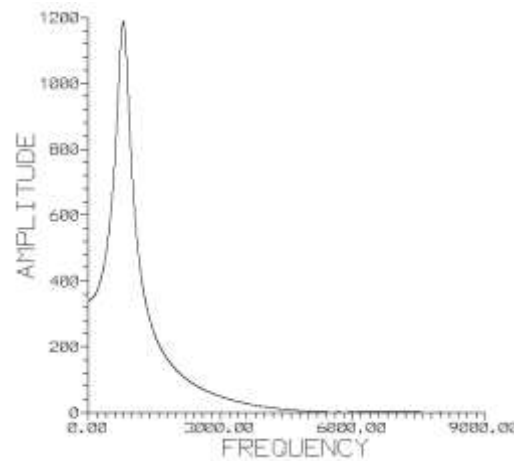
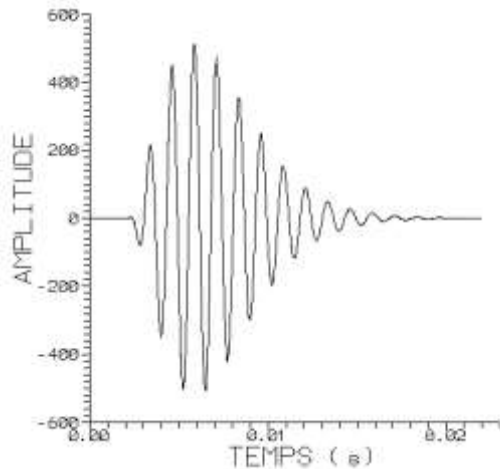
Analysis/synthesis of a signal model

■ Analysis/synthesis by « FOF »

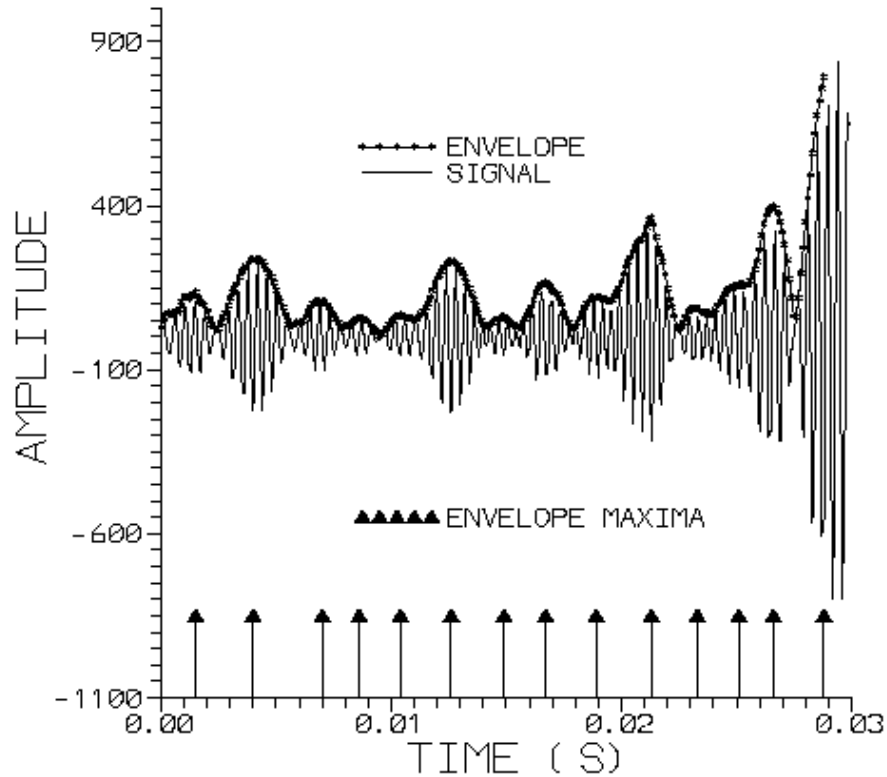
$$f(t) = \Lambda(t) \sin(\omega_c t + \phi)$$

$$\Lambda(t) = \begin{cases} 0 & \text{si } t \leq 0 \\ \frac{1}{2}A(1 - \cos(\beta t))e^{-\alpha t} & \text{si } 0 < t \leq \pi/\beta \\ Ae^{-\alpha t} & \text{si } t > \pi/\beta \end{cases}$$

● *5 formants*



Analysis/synthesis with FOF (2)



■ Exemple (after Potard, Rodet, Ircam)



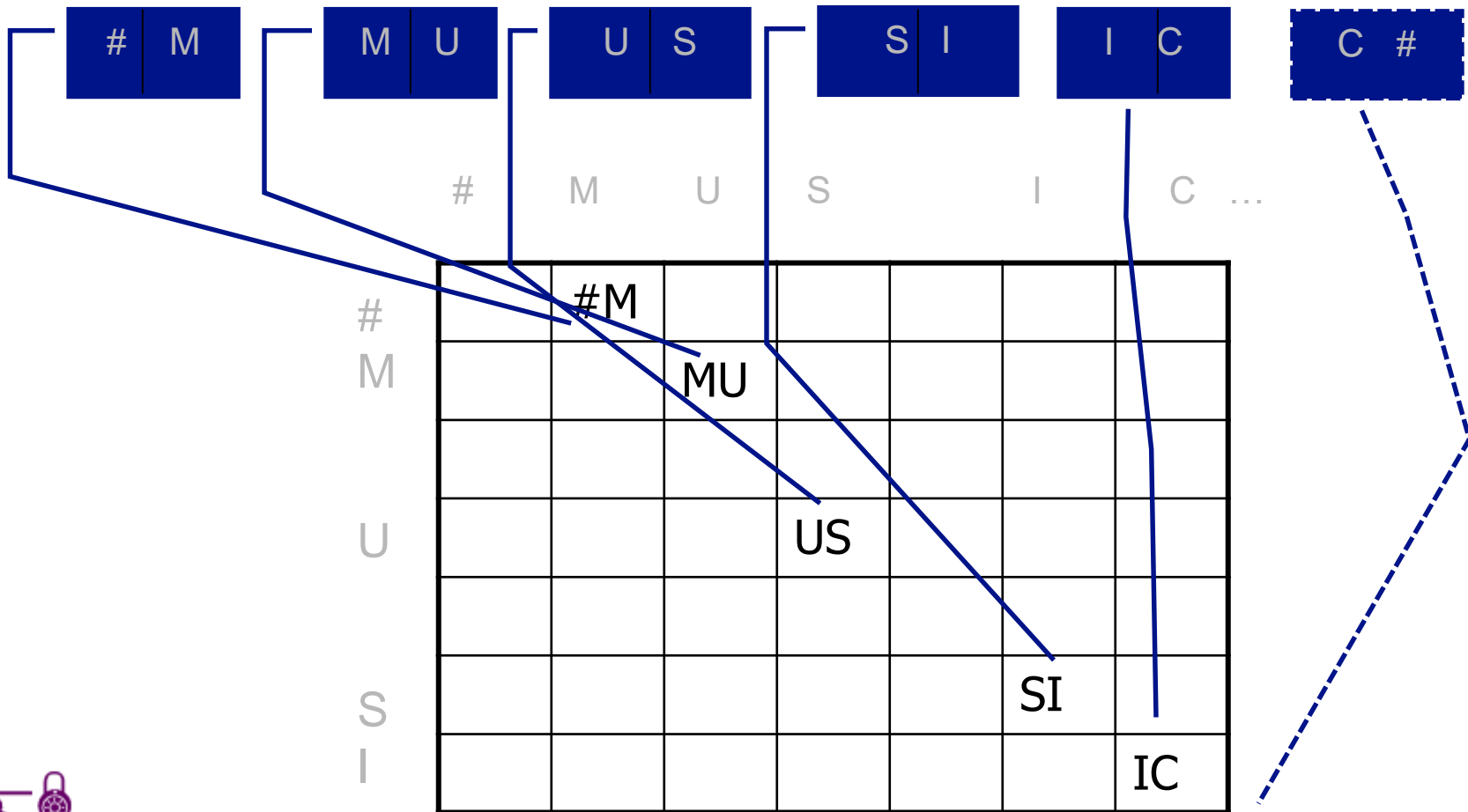
Wavetable Synthesis

- Also called by « sampling »
- Digital sound dictionaries
 - 1 or several periods
 - Real instruments or/and classic functions
- Widely spread (especially low end sound cards)
- Needs for :
 - Pitch shifting, re-sampling
 - Loops handling



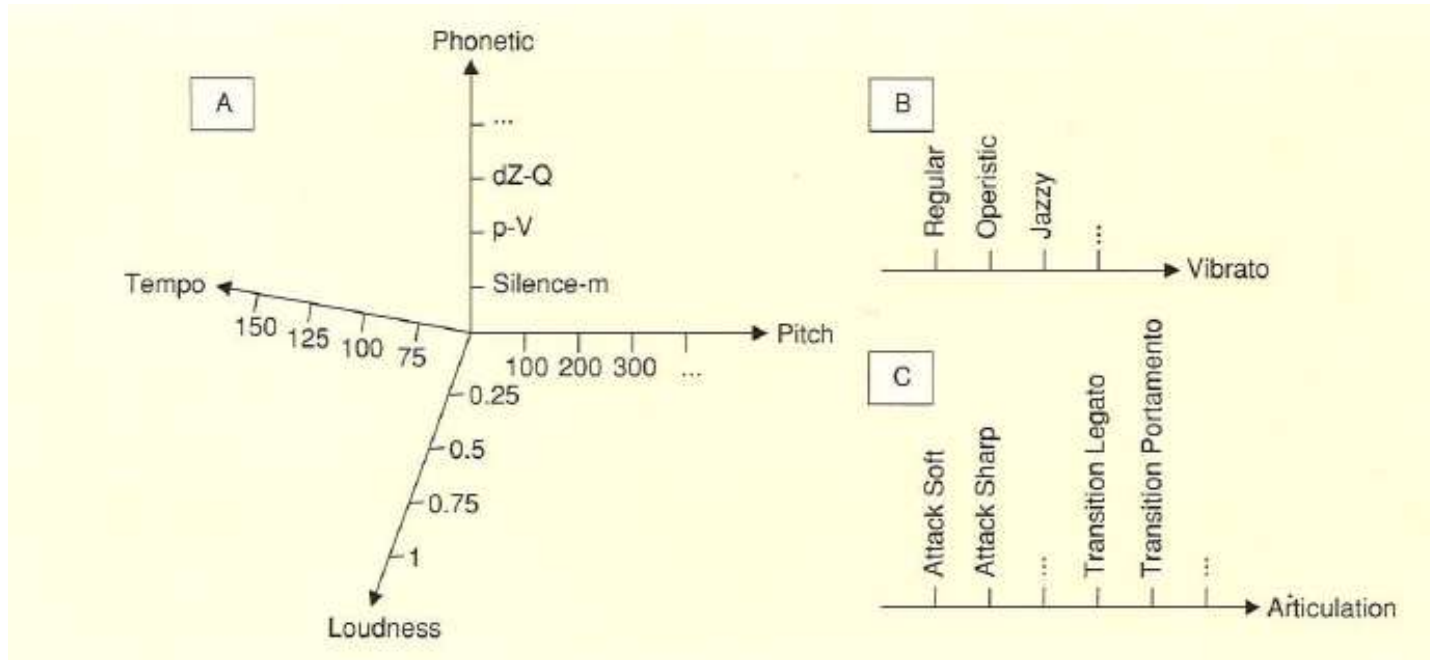
Synthesis by concatenation

- Inspired from speech synthesis by concatenation



Concatenative synthesis

- A bit more complex for singing voice
- Parametrisation of a singing voice space (d'après Bonada & al.)

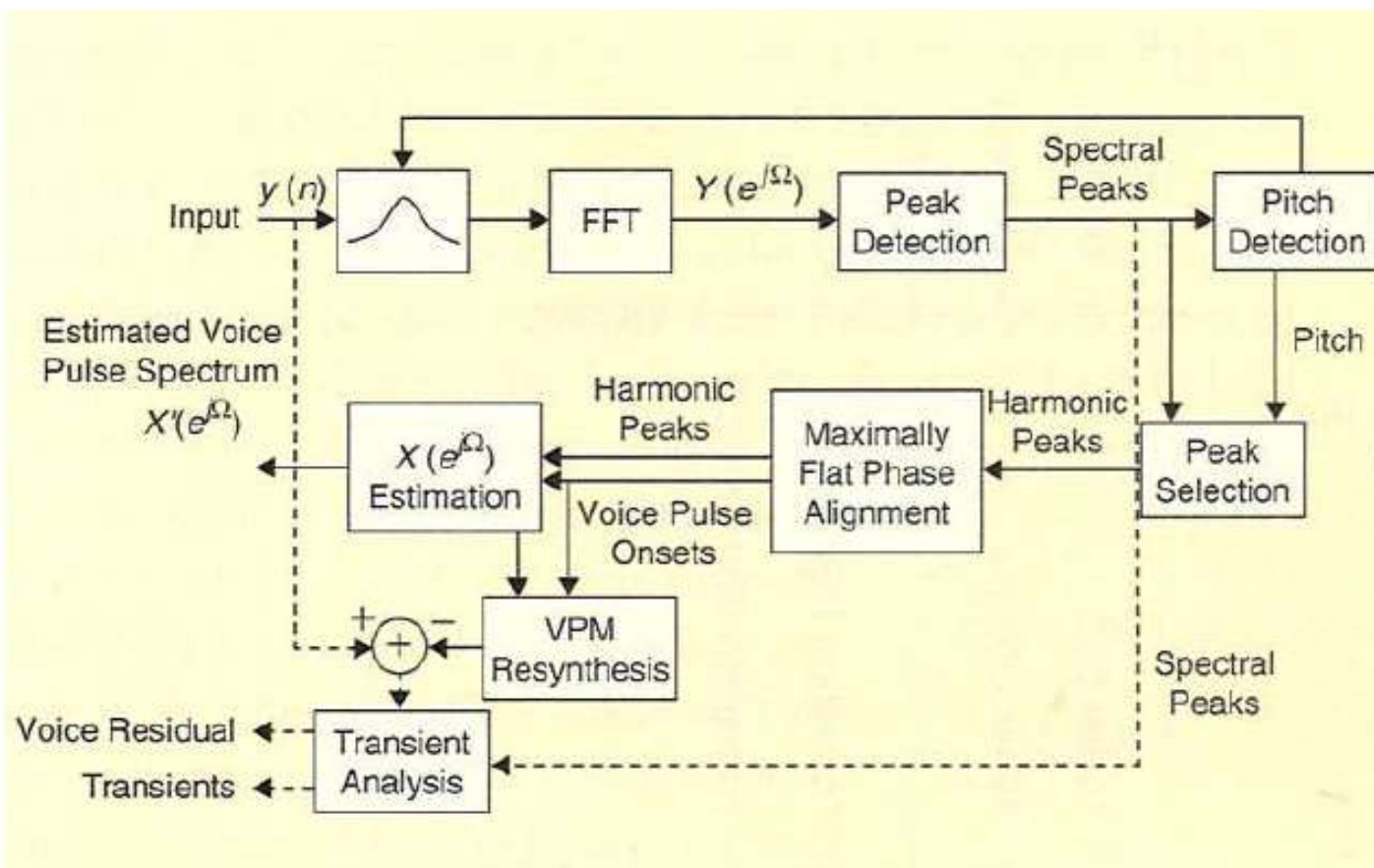


J. Bonada and X. Serra, "Synthesis of the Singing Voice by Performance Sampling and Spectral Models," in *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 67-79, March 2007, doi: 10.1109/MSP.2007.323266.



Concatenative synthesis

■ Analysis scheme for pre-recorded segments modification

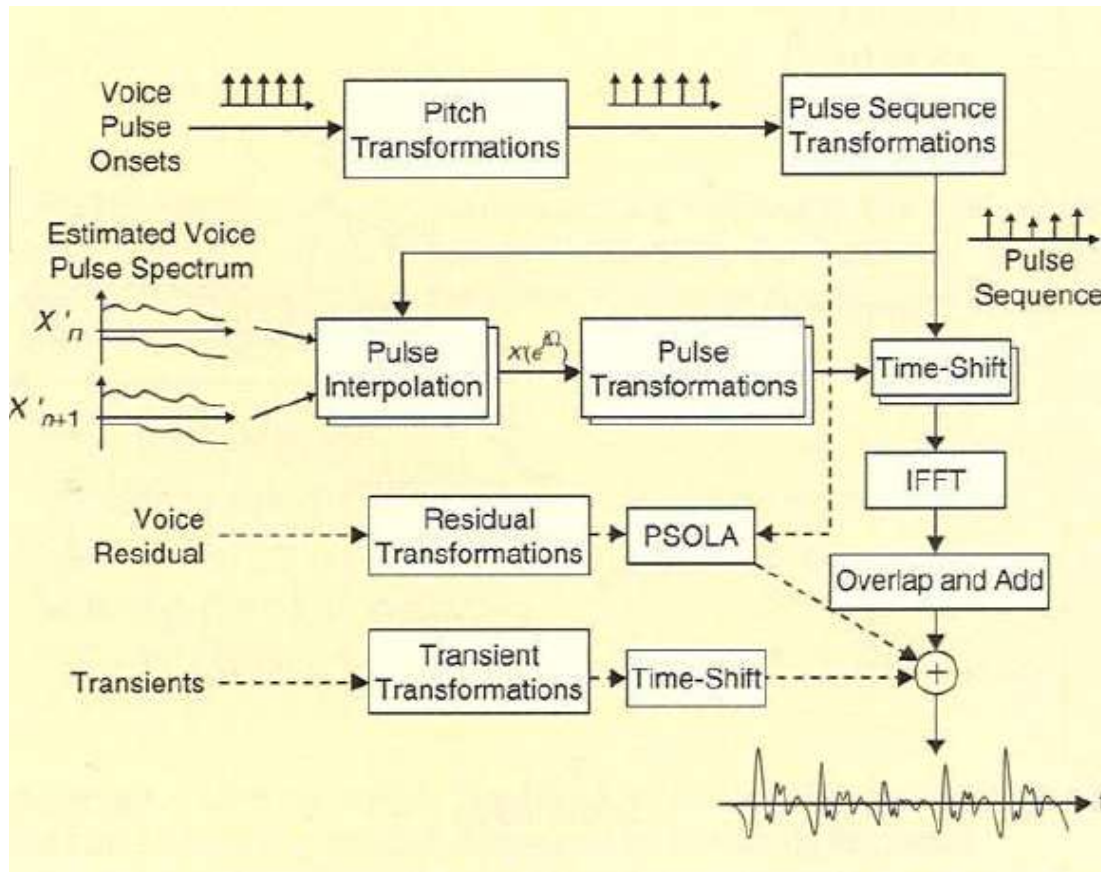


J. Bonada and X. Serra, "Synthesis of the Singing Voice by Performance Sampling and Spectral Models," in *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 67-79, March 2007, doi: 10.1109/MSP.2007.323266.



Concatenative synthesis

■ Synthesis scheme for pre-recorded segments modification



J. Bonada and X. Serra, "Synthesis of the Singing Voice by Performance Sampling and Spectral Models," in *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 67-79, March 2007, doi: 10.1109/MSP.2007.323266.



Concatenative synthesis

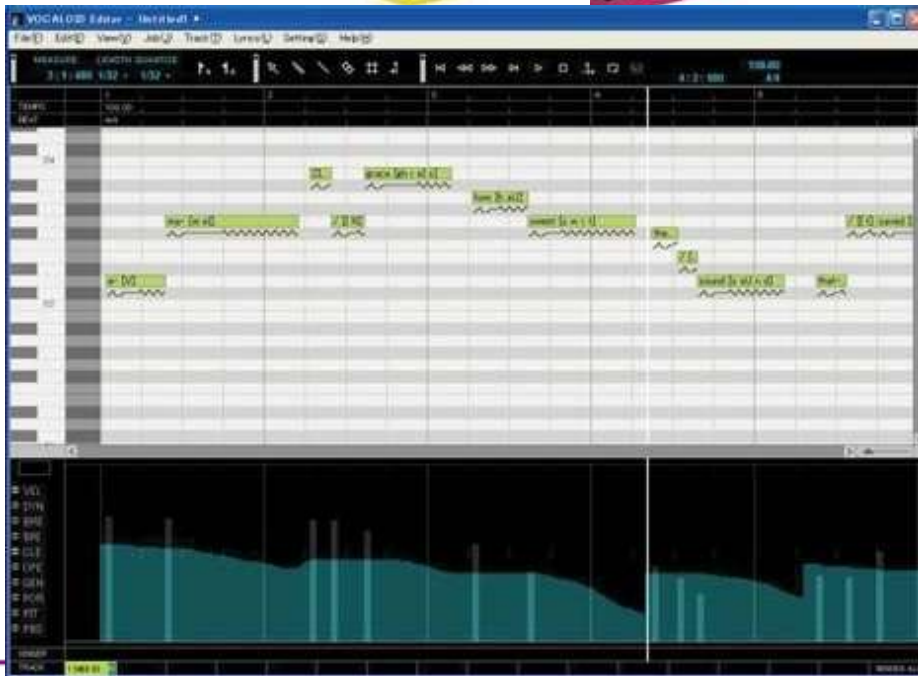
■ Sound example from the original VOCALOID (Yamaha, 2007)



Sweet dreams



Knightmare



Other examples in Japanese on the net:

[Hatsune Miku](#)

Other recent sound examples (VOCALOID6) with machine learning:

<https://www.vocaloid.com/en/vocaloid6/>

Images from: Kenmochi, H., Ohshita, H.
VOCALOID - commercial singing synthesizer based on sample concatenation. Proc. Interspeech 2007, 4009-4010



Analysis/synthesis of a production model

“using a model, source elements and operators”

■ FM synthesis (frequency modulation)

- Historically very popular
- ..and still use
- Principle:
 - Inspired from the transmission of Hertzian waves but with carrier and modulation frequency of the same order of magnitude
 - Phase modulation and instantaneous frequency

$$x(t) = A \sin(\phi(t))$$

$$\phi(t) = 2\pi f_p t + I \sin(2\pi f_m t)$$

$$f_i(t) = f_p + I f_m \cos(2\pi f_m t) = \frac{1}{2\pi} \frac{\partial \phi}{\partial t}$$



FM synthesis (2)

■ Temporal signal

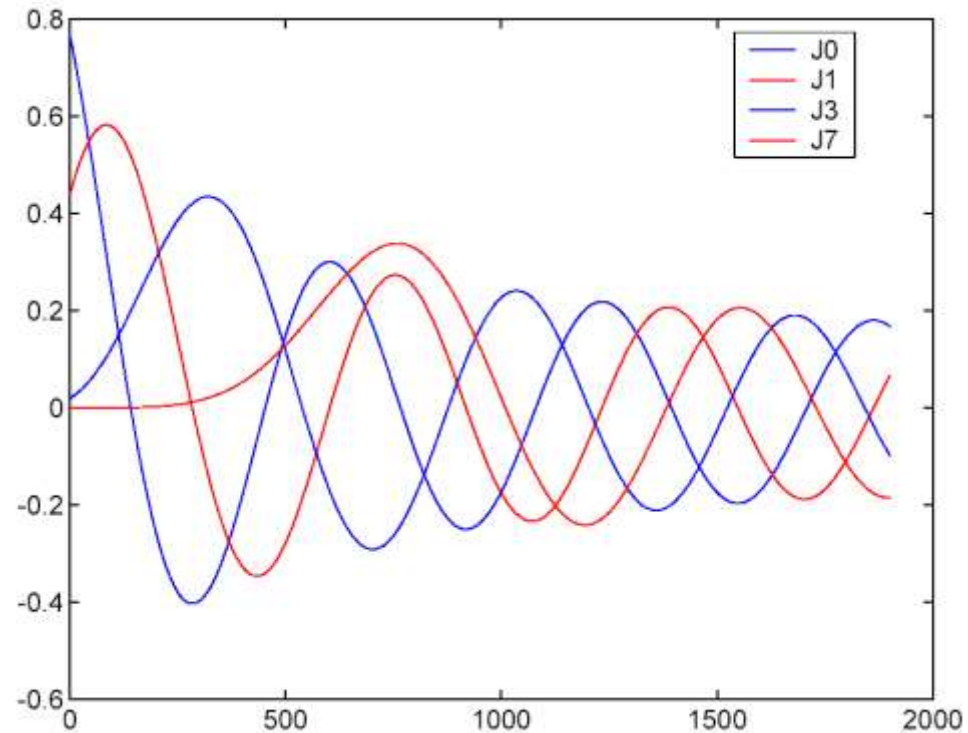
$$\begin{aligned}x(t) &= A \sin(\phi(t)) \\ &= AJ_0(I) \sin(2\pi f_p t) \\ &\quad + A \sum_{n=1}^{+\infty} J_n(I) \sin(2\pi(f_p + n f_m)t) \\ &\quad + A \sum_{n=1}^{+\infty} J_n(I) (-1)^n \sin(2\pi(f_p - n f_m)t)\end{aligned}$$



FM synthesis (3)

■ Bessel function of first kind

- Damped and delayed waveforms
- Bandwidth larger when I increases
- Spectrum variation rather unpredictable when I is modulated

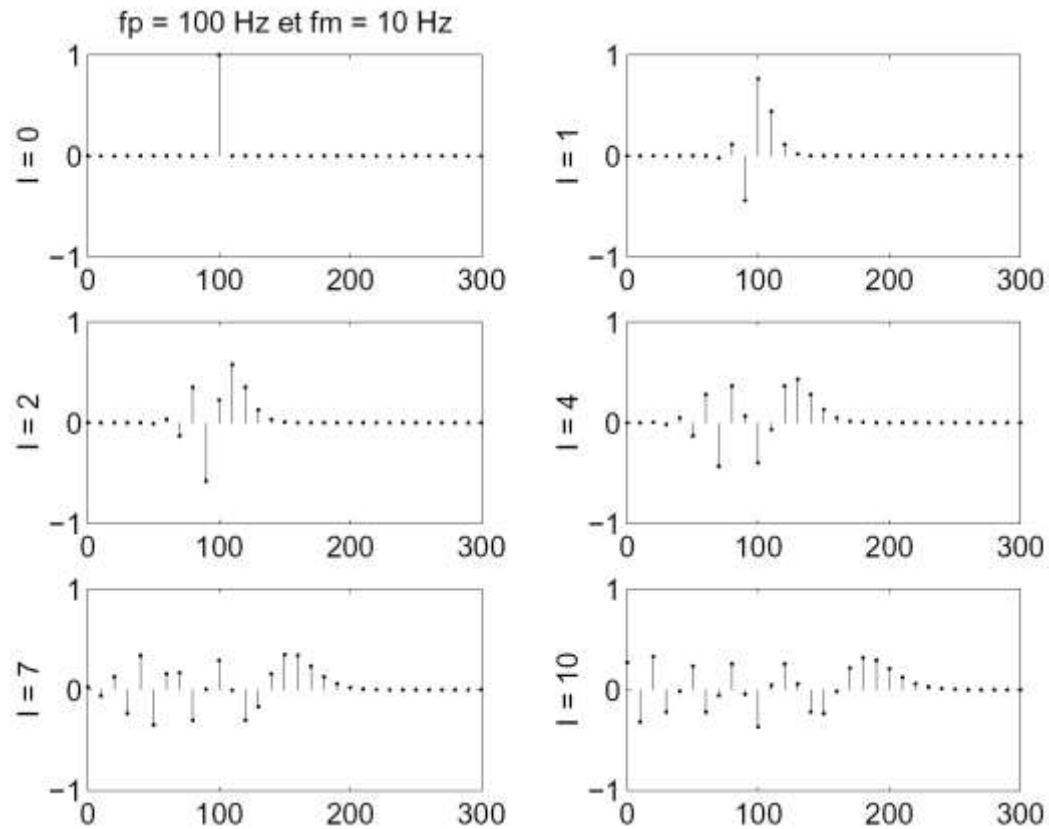


Modulation index I



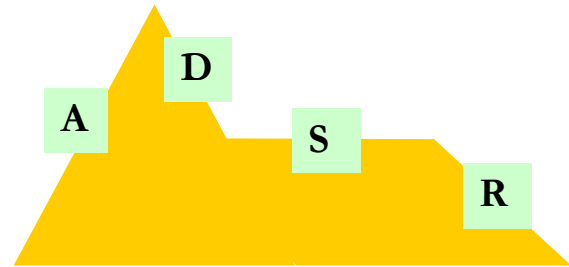
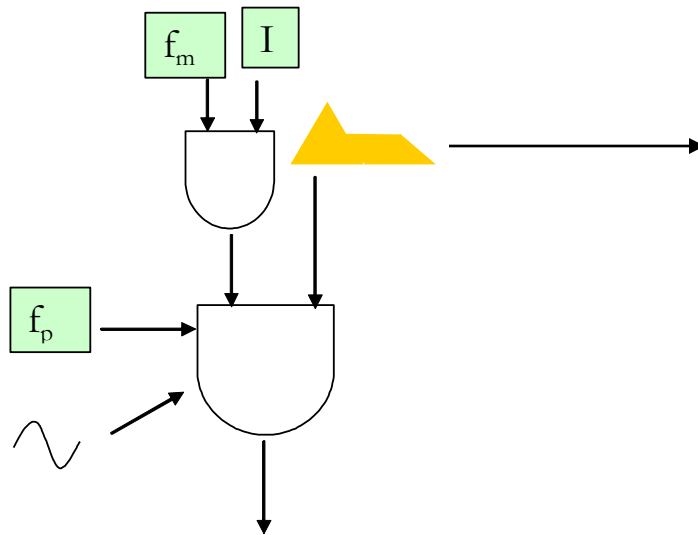
FM synthesis (4)

■ Spectrum variation with I



FM synthesis (5)

■ Functional scheme + time domain envelope

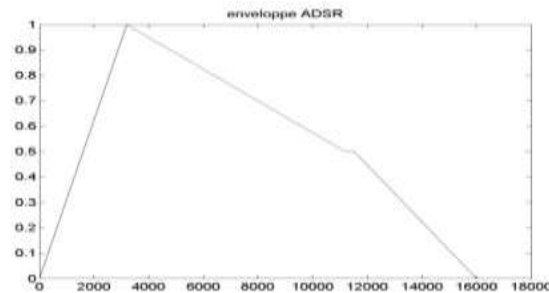


ADSR envelope model



Some simple examples

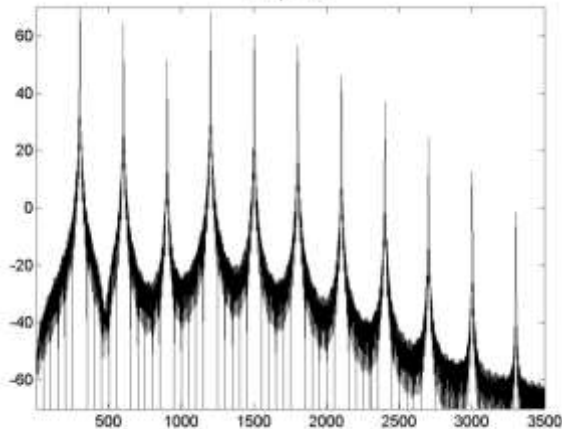
■ *Envelope:*



Fp:fm = 1:1 Son cuivré



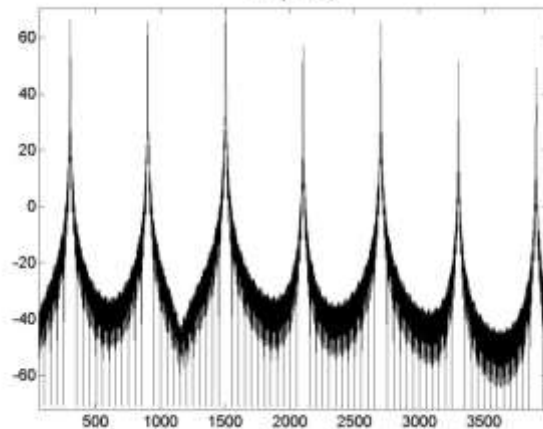
fm:fp = 1:1



Fp:fm = 1:2 Son boisé



fm:fp = 1:2

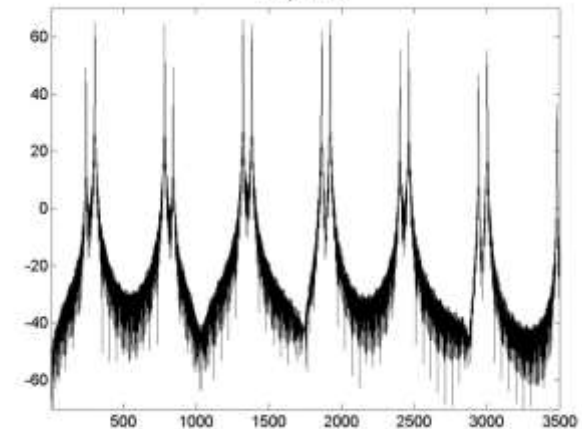


Fp:fm = 1:1.8

Son inharmonique



fm:fp = 1:1.8



More elaborated examples

(from *dartmouth music college – former site*)

■ A kind of Bell sound



- F_p (carrying): 100 Hz, F_m (modulation): 280 Hz, I (modulation index): 6.0 $\rightarrow 0$

■ A kind of Marimba sound:



- F_p : 250 Hz, F_m : 175 Hz, I : 1.5 $\rightarrow 0$

■ A kind of trumpet sound

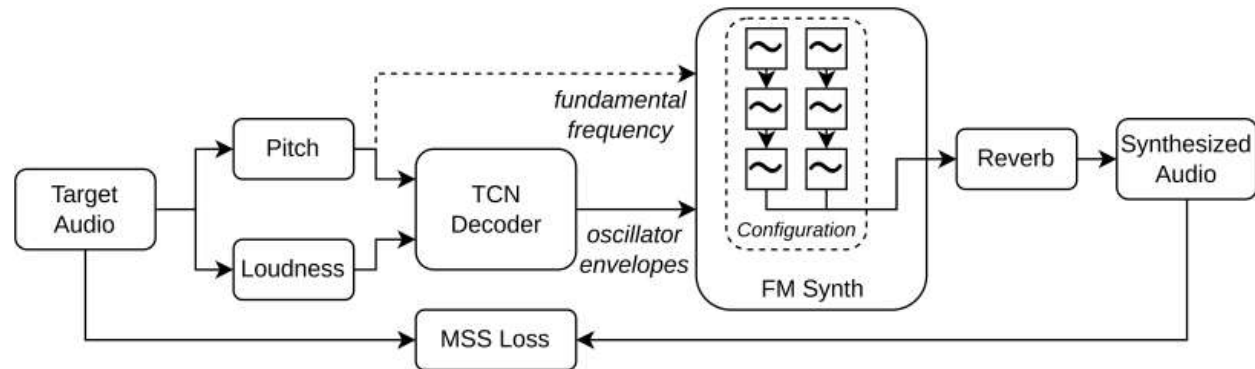


- F_p : 700 Hz, F_m : 700 Hz, I : 5.0 $\rightarrow 0$



FM synthesis : some conclusions

- (was) very popular
- Very low cost
 - With lots of different timbre
 - Easy to generate sounds
- But very difficult to find the synthesis parameters (highly non linear – required a lot of «expertise »)
- Recent work to estimate synthesis parameters with deep learning



Caspe, F.S., Mcpherson, A.P., & Sandler, M.B. (2022). DDX7: Differentiable FM Synthesis of Musical Instrument Sounds. ArXiv, abs/2208.06169.



Physical model

" aims at reproducing the real behavior of the instrument through a physical model ".

■ Many approaches... more or less complex

- Karplus-Strong model
- Waveguide models
- Based on fundamental principle of physics
- ..



Physical model

" aims at reproducing the real behavior of the instrument through a physical model ".

■ Many approaches... more or less complex

- Karplus-Strong model
- Waveguide models
- Based on fundamental principle of physics



Physical model

" aims at reproducing the real behavior of the instrument through a physical model "

■ Based on fundamental principal of physics

- Mass conservation
- Quantity of movement conservation
- Moments conservation
- 1st principle of thermodynamics (energie)
- 2nd principe of thermodynamics (entropy)

➔ **movement equations**

■ Applicable to physical systems of any geometry and anly material.

■ Additional constraints/rules use to take into acount the specific nature of the object.

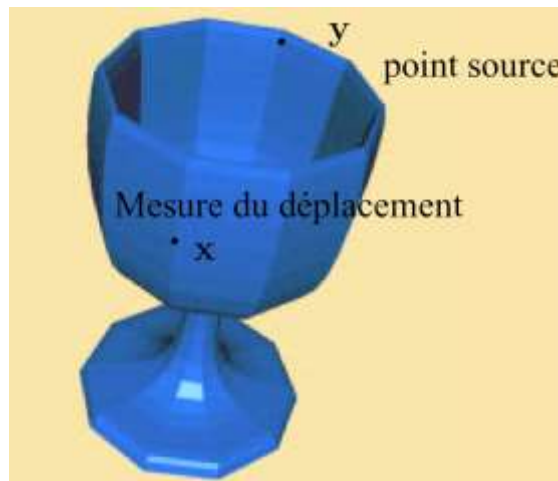


A simulation example (from Bensoam, IRCAM)

- Discretisation in time of the equations
- Space discretisation (i.e. shape of the instrument)
- Simulation experiment

Force $\delta(t)$ applied in $y \Rightarrow$ Response $P(x,y;t)$

$$U(x,t) = P(x,y) * \delta(t)$$

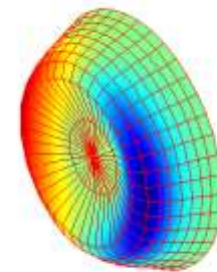
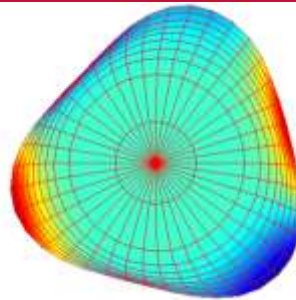
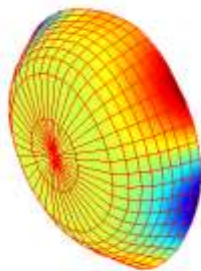
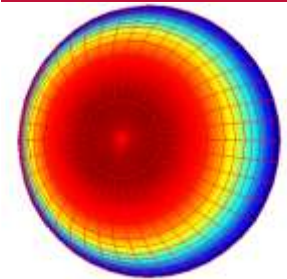
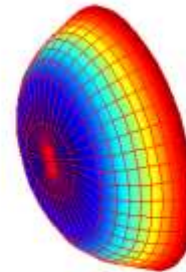
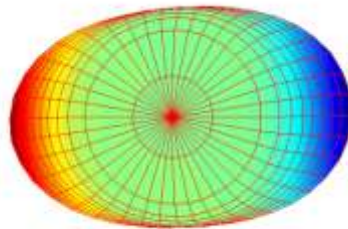
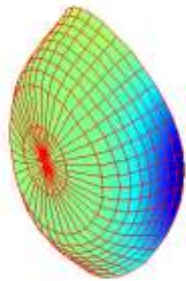


Sound synthesis

(from Bensoam, IRCAM)



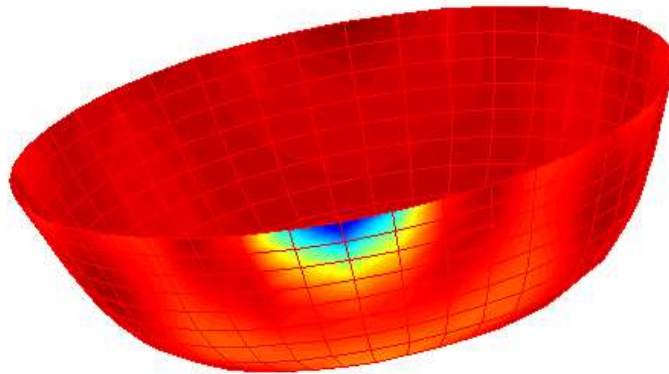
- Simulation of vibration modes 1 to 7 of a Tibetan bowl



Sound synthesis

(from Bensoam, IRCAM)

■ Tibetan bowl hit on the top



 Movement

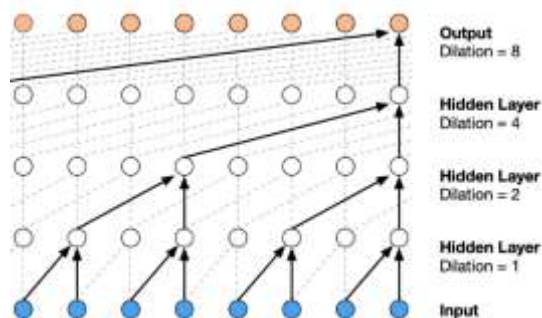
 Acceleration



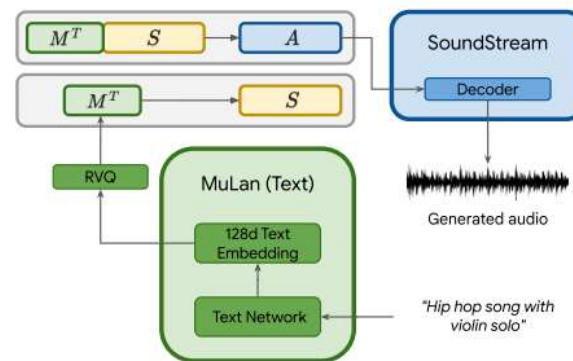
Deep neural audio synthesis

- Machine-learning based models "uses large amount of data and machine learning to generate sounds"
 - A rapid growth and adoption of deep neural networks for audio synthesis

From wavenet (2016)
(autoregressive model)



MusicLM (2023)
(Generating Music from Text)



Deep neural audio synthesis

a snapshot, from J. Nistal, "Exploring Generative Adversarial Networks for Controllable Musical Audio Synthesis, PhD Thesis, IP Paris, 2022

Arch.	Name	Audio representation	Data	Conditioning	
NAM	waveNet	van den Oord et al., 2016a	waveform	speech, piano	speaker ID, text
	Universal music Translation	Mor et al., 2018	waveform	classical music	-
	Hierarchical waveNet	Dieleman et al., 2018	waveform	piano music	-
	SampleRNN	Mehri et al., 2017	waveform	speech, piano music	-
	MelNet	Vasquez and Lewis, 2019	mag. spec.	speech, piano music	speaker ID, text
	wavenetAE	Engel et al., 2017	waveform	tonal sounds	pitch
	sparse Transformer	Child et al., 2019	waveform	piano music	-
NFs	Parallel waveNet	van den Oord et al., 2018a	waveform	speech	text, pitch
	ClariNet	Ping et al., 2018	waveform	speech	text
	FlowwaveNet	Kim et al., 2018	waveform	speech	text, Mel spec.
	waveGlow	Prenger et al., 2018	waveform	speech	text, Mel spec.
	waveFlow	Ping et al., 2020	waveform	speech	text, Mel spec.
	Blow	Serrà et al., 2019	waveform	speech	speaker ID
VAEs	Planet Drums	Aouameur et al., 2019	Mel-scaled mag. spec.	drums	instrument ID
	Jukebox	Dhariwal et al., 2020	waveform	music	artist & genre ID, lyrics
	NOTONO	Bazin et al., 2020	mag. & IF	tonal instruments	pitch
	FlowSynth	Esling et al., 2019	mag.	synth. sounds	semantic tags
	Neural Granular Sound Synth.	Bitton et al., 2020	waveform	orchestral drums, animals	pitch, instrument ID
GANs	WaveGAN	Donahue et al., 2019	waveform	speech, drums, piano, birds	-
	GANSynth	Engel et al., 2019	mag. & IF	tonal instruments	pitch ID
	MelGAN	Kumar et al., 2019	mag. spec.	speech, music	Mel-scaled spec. text
	GAN-TTS	Binkowski et al., 2020	waveform	speech	pitch, text, speaker ID



Wavnet

a generative model, directly from the audio waveform

- The joint probability of a waveform $\mathbf{x} = \{x_1, \dots, x_T\}$ is factorised as a product of conditional probabilities :

$$p(\mathbf{x}) = \prod_{t=1}^T p(x_t | x_1, \dots, x_{t-1})$$

- the conditional probability distribution is modelled by a stack of convolutional layers;
- Output of the model: has the same time dimensionality as the input (no pooling)
- Output: a categorical distribution over the next value x_t with a softmax layer - optimized to maximize the log-likelihood of the data w.r.t. the parameters.

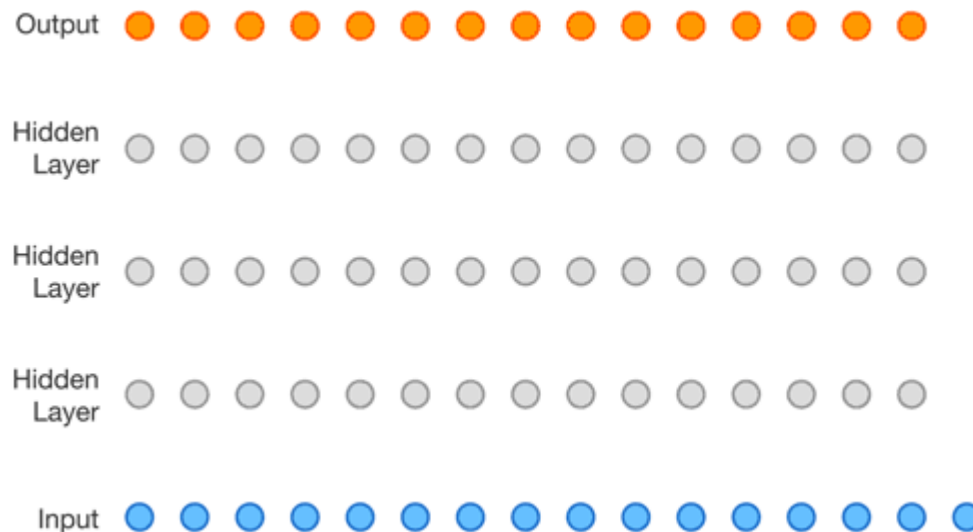
Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio (cite arxiv:1609.03499)



Wavnet

a generative model, directly from the audio waveform

- Dilated causal convolutions (the main ingredient!).
 - Classic causal convolutions needs many layers to increase the receptive fields (RF)
 - Dilated causal convolutions greatly increase RF



Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio (cite arxiv:1609.03499)



Wavnet

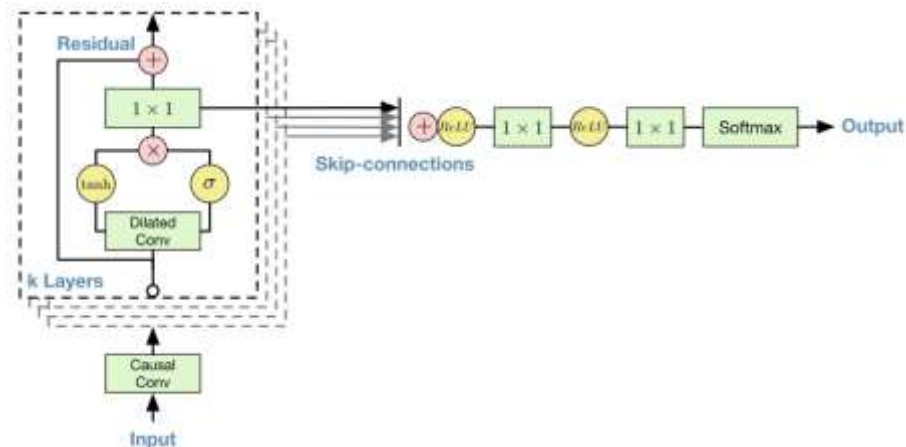
a generative model, directly from the audio waveform

- Condition distributions $p(x_t|x_1, \dots, x_{t-1})$ modelled using softmax distributions
- Use of mu-law to limit the number of “categories (amplitude values):

$$f(x_t) = \text{sign}(x_t) \frac{\ln(1 + \mu |x_t|)}{\ln(1 + \mu)}$$

- Use of Gated recurrent units $\mathbf{z} = \tanh(W_{f,k} * \mathbf{x}) \odot \sigma(W_{g,k} * \mathbf{x})$

- .. and residual and skip connections



Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio (cite arxiv:1609.03499)



Wavnet : sound examples

(from <https://www.deepmind.com/blog/wavenet-a-generative-model-for-raw-audio>)

- Speech ... but also music (no conditioning)



■ But it is also possible to use conditions :

$$p(\mathbf{x} | \mathbf{h}) = \prod_{t=1}^T p(x_t | x_1, \dots, x_{t-1}, \mathbf{h})$$

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k}^T \mathbf{h}) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k}^T \mathbf{h})$$

- Speech (with condition on the text)



Wavnet and other neural autoregressive models

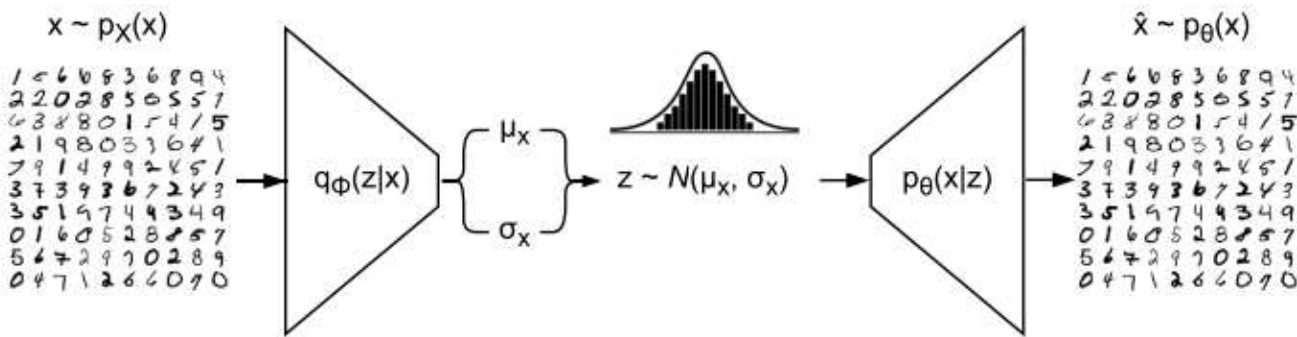
- Wavnet remains complex (sample is generated one at a time)
- Other neural autoregressive models

Arch.	Name	Audio representation	Data	Conditioning	
NAM	waveNet	van den Oord et al., 2016a	waveform	speech, piano	speaker ID, text
	Universal music Translation	Mor et al., 2018	waveform	classical music	-
	Hierarchical waveNet	Dieleman et al., 2018	waveform	piano music	-
	SampleRNN	Mehri et al., 2017	waveform	speech, piano music	-
	MelNet	Vasquez and Lewis, 2019	mag. spec.	speech, piano music	speaker ID, text
	wavenetAE	Engel et al., 2017	waveform	tonal sounds	pitch
	sparse Transformer	Child et al., 2019	waveform	piano music	-



Variational AutoEncoders

Schematic principle of Variational Autoencoders (VAEs)



The encoder $q_\phi(z|x)$ approximates the true posterior distribution $q_\theta(z|x)$

The decoder $p_\theta(z|x)$ generates an approximation \hat{x} from the encoding

Main idea of variational inference: :

- The complete model $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, but the data follows complex distributions
- Exploit an approximate of the true posterior: $q_\phi(\mathbf{z}|\mathbf{x})$
- Variational inference: minimizing the difference between the approximation and the true density:

$$q_\phi^*(\mathbf{z}|\mathbf{x}) = \operatorname{argmin}_{q_\phi(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}} D_{\text{KL}}[q_\phi(\mathbf{z}|\mathbf{x})|p(\mathbf{z}|\mathbf{x})]$$



Variational AutoEncoders

$$q_{\phi}^*(\mathbf{z}|\mathbf{x}) = \operatorname{argmin}_{q_{\phi}(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}} D_{\text{KL}}[q_{\phi}(\mathbf{z}|\mathbf{x})|p(\mathbf{z}|\mathbf{x})]$$

- This can be further expressed as :

$$\log p_{\theta}(\mathbf{x}) = D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x})|p_{\theta}(\mathbf{z}|\mathbf{x})) + \mathcal{L}(\phi, \theta, \mathbf{x})$$

- It describes the quantity to model $\log p_{\theta}(\mathbf{x})$ minus the error we make by using an approximate q instead of the true p .
- We can maximize the **Evidenced Lower Bound (ELBO)**

$$\mathcal{L}(\phi, \theta, \mathbf{x}) = -D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}(\log p_{\theta}(\mathbf{x}|\mathbf{z}))$$

Kingma et Welling, « An Introduction to Variational Autoencoders », *Foundations and Trends in Machine Learning*, vol. 12, n° 4, 2019, p. 307–392

K. Sachdeva: “Evidence Lower Bound (ELBO) - CLEARLY EXPLAINED!”
<https://www.youtube.com/watch?v=IXsA5Rpp25w>



Variational AutoEncoders in Audio/music

Many examples

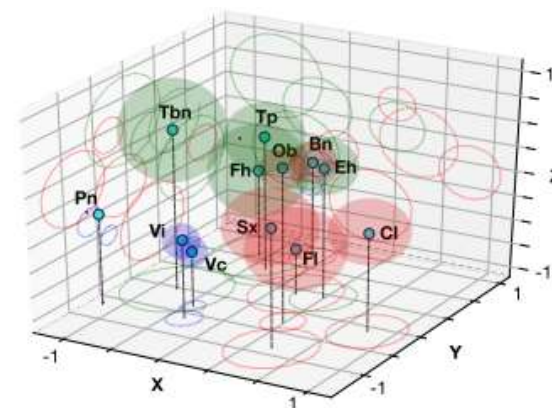
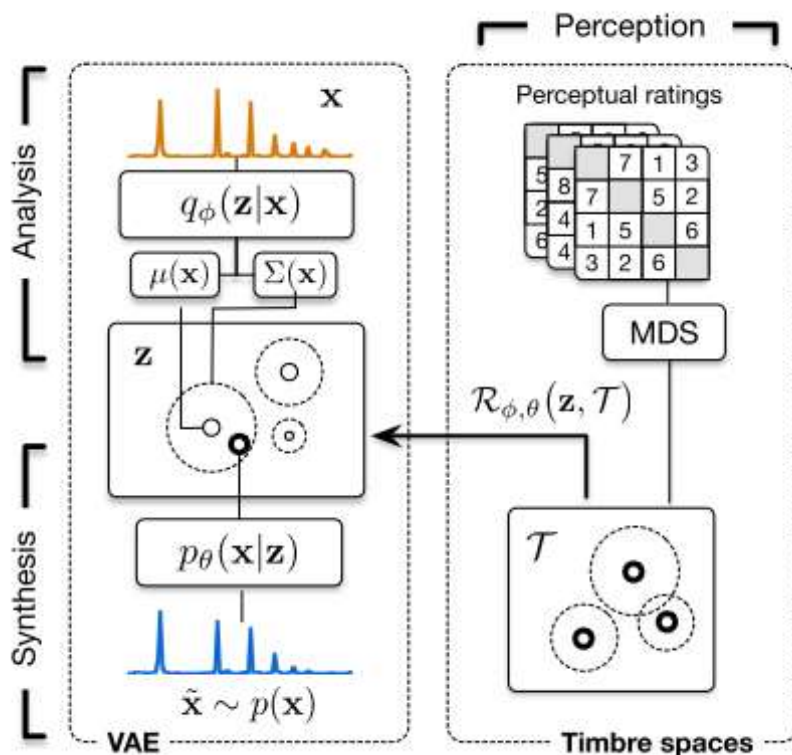
Arch.	Name	Audio representation	Data	Conditioning
VAEs	Planet Drums	Aouameur et al., 2019	Mel-scaled mag. spec.	drums instrument ID
	Jukebox	Dhariwal et al., 2020	waveform	music artist & genre ID lyrics
	NOTONO	Bazin et al., 2020	mag. & IF	tonal instruments pitch
	FlowSynth	Esling et al., 2019	mag.	synth. sounds semantic tags
	Neural Granular Sound Synth.	Bitton et al., 2020	waveform	orchestral drums animals pitch instrument ID

J. Nistal, PhD thesis, 2022



Variational AutoEncoders in Audio/music

Regularizing the latent space with timbre spaces (perception)



Multi-dimensional scaling (MDS)



P. Esling, A. Chemla-Romeu-Santos, A. Bitton, Bridging Audio Analysis, Perception and Synthesis with Perceptually-regularized Variational Timbre Spaces, in Proc. of ISMIR » 2018 »

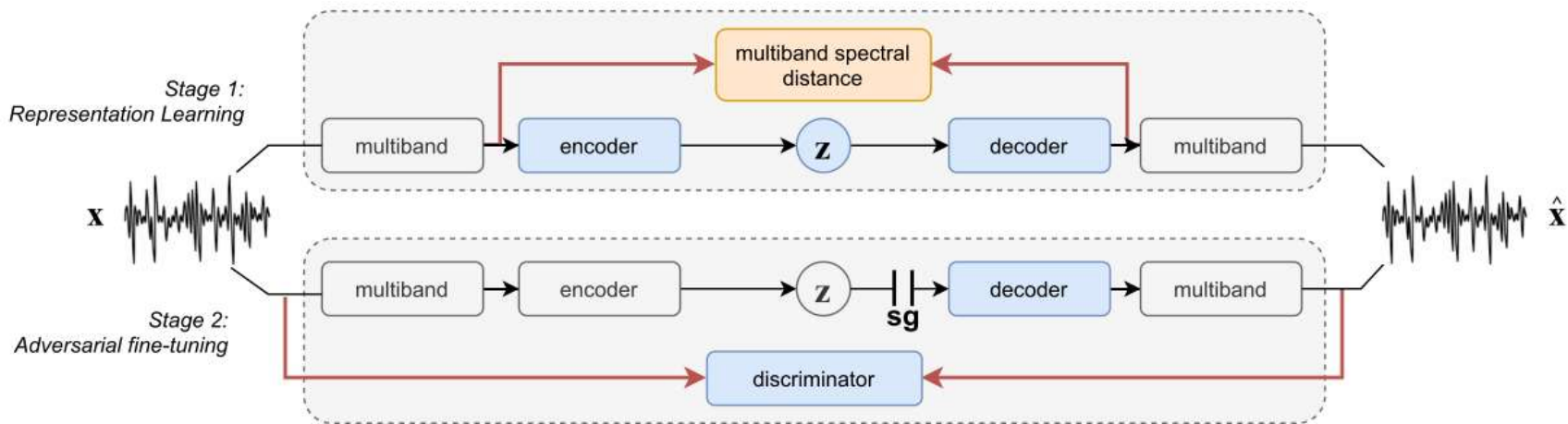


Variational AutoEncoders in Audio/music

Extensions

RAVE: Realtime Audio Variational autoEncoder

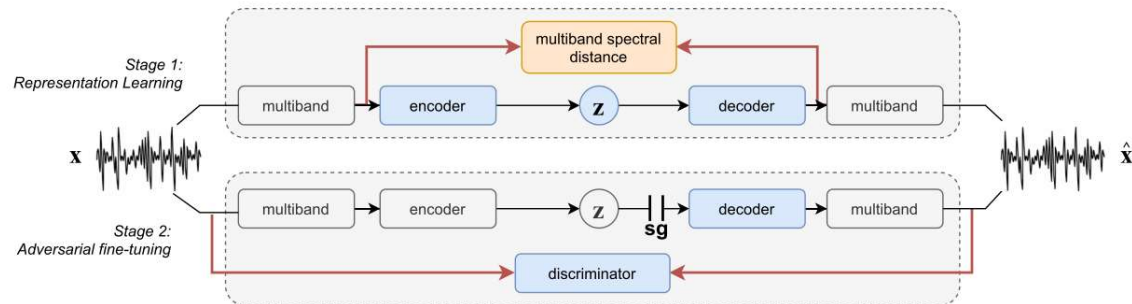
- Based on a two stage training:
 1. representation learning with VAEs (stage 1)
 2. adversarial fine tuning (stage 2)



A. Caillon, Antoine; P. Esling. "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis." *ArXiv abs/2111.05011* (2021)

Variational AutoEncoders in Audio/music

RAVE : some details



- The multispectral loss (from Engel2019 (DDSP))

$$S(\mathbf{x}, \mathbf{y}) = \sum_{n \in \mathcal{N}} \left[\frac{\|\text{STFT}_n(\mathbf{x}) - \text{STFT}_n(\mathbf{y})\|_F}{\|\text{STFT}_n(\mathbf{x})\|_F} + \log(\|\text{STFT}_n(\mathbf{x}) - \text{STFT}_n(\mathbf{y})\|_1) \right]$$

- Latent representation compactness
 - To avoid *posterior* collapse (e.g situation where **the learned latent space is ignored**)
 - Based on variance normalisation, rank estimation (using SVD)



A. Caillon, Antoine; P. Esling. "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis." ArXiv abs/2111.05011 (2021)



Variational AutoEncoders in Audio/music

RAVE : some results

- Evaluation (in 2021)

Model	MOS	95% CI	Training time	Parameter count
Ground truth	4.21	± 0.04	-	-
NSynth	2.68	± 0.04	~ 13 days	64.7M
SING	1.15	± 0.02	~ 5 days	80.8M
RAVE (Ours)	3.01	± 0.05	~ 7 days	17.6M

- Synthesis examples:

- Timbre transfer (model trained on speech, input :violin)



Violin input



Output

- Darbouka synthesis:

- Reconstruction



Original



Reconstructed

- Unconditional generation



unconditioned



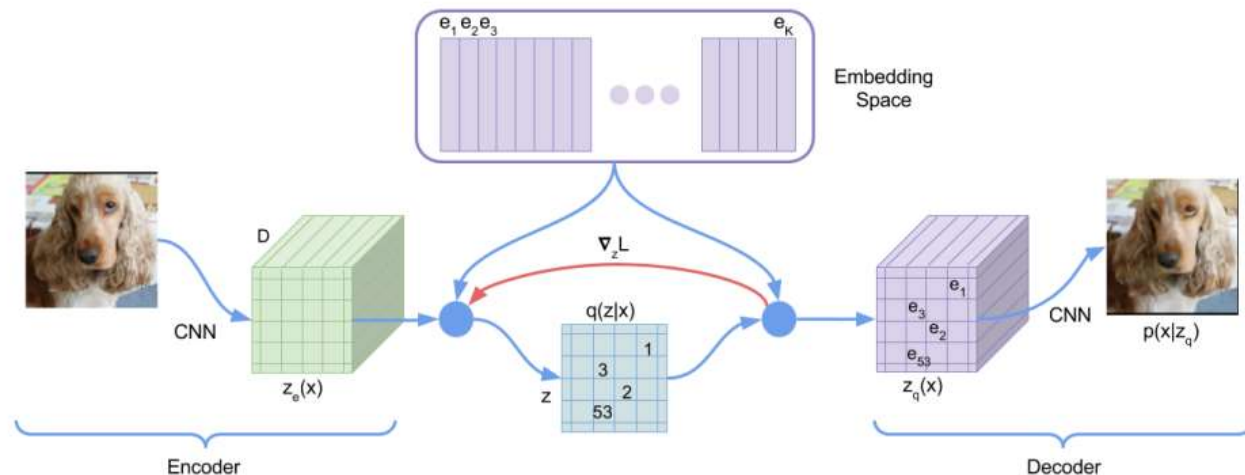
A. Caillon, Antoine; P. Esling. "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis." *ArXiv abs/2111.05011* (2021)



Vector-Quantized Variational AutoEncoders (VQ-VAEs)

- Combines VAEs with Vector quantization
- Helps to avoid *posterior* collapse of VAEs
- Offers the flexibility of a **discrete** neural representation

■ Main principle



A. van den Oord, O. Vinyals, K. Kavukcuoglu.. Neural discrete representation learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). 2017

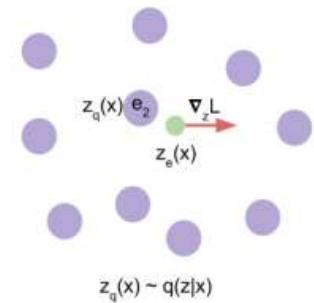
Vector-Quantized Variational AutoEncoders (VQ-VAEs)

■ Discrete latent representation

- The discrete latent variables are obtained by nearest neighbour look-up

$$q(z = k|x) = \begin{cases} 1 & \text{for } k = \operatorname{argmin}_j \|z_e(x) - e_j\|_2, \\ 0 & \text{otherwise} \end{cases}$$

$$z_q(x) = e_k, \quad \text{where } k = \operatorname{argmin}_j \|z_e(x) - e_j\|_2$$



A.van den Oord, O. Vinyals, K. Kavukcuoglu.. Neural discrete representation learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). 2017



Vector-Quantized Variational AutoEncoders (VQ-VAEs)

■ Learning

- A loss function with three components
 1. A reconstruction loss (or data term)
 2. A dictionary learning term (VQ):
 3. A commitment loss (to force a joint learning of encoder and dictionary)

$$L = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta \|z_e(x) - \text{sg}[e]\|_2^2$$



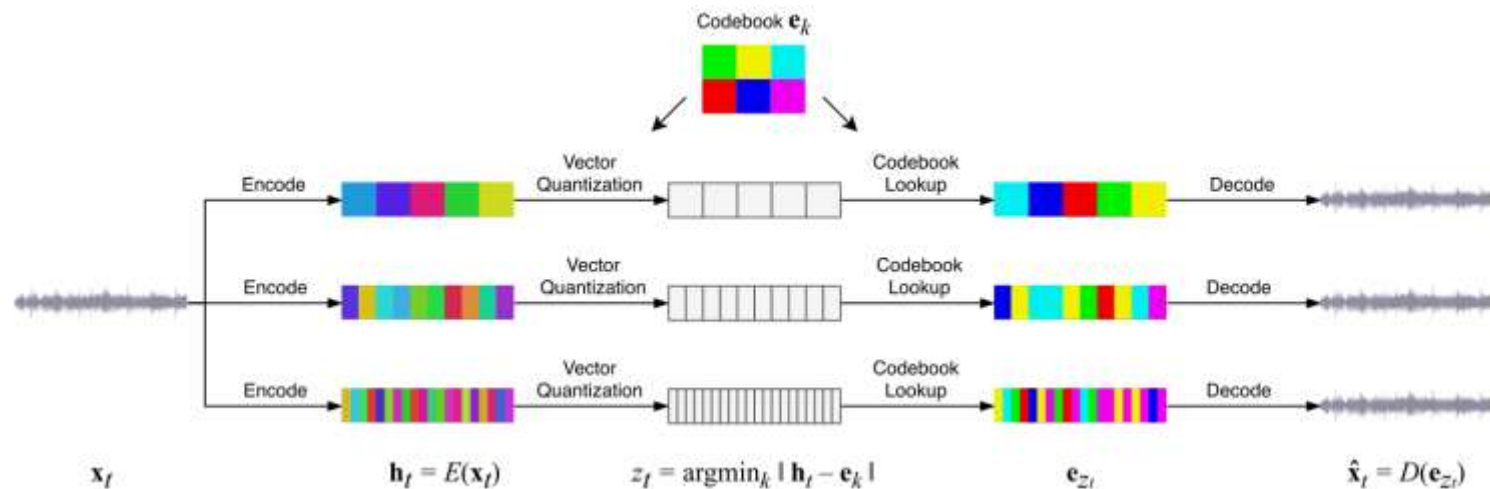
A. van den Oord, O. Vinyals, K. Kavukcuoglu.. Neural discrete representation learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). 2017



VQ-VAEs in Audio and Music

An example with Jukebox

- Based on hierarchical VQ-VAE (VQ-VAE2), trained with an additional spectral loss
- Combined with sparse transformers for learning the latent prior for generation



Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. In Advances in Neural Information Processing Systems, 2019.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509, 2019.

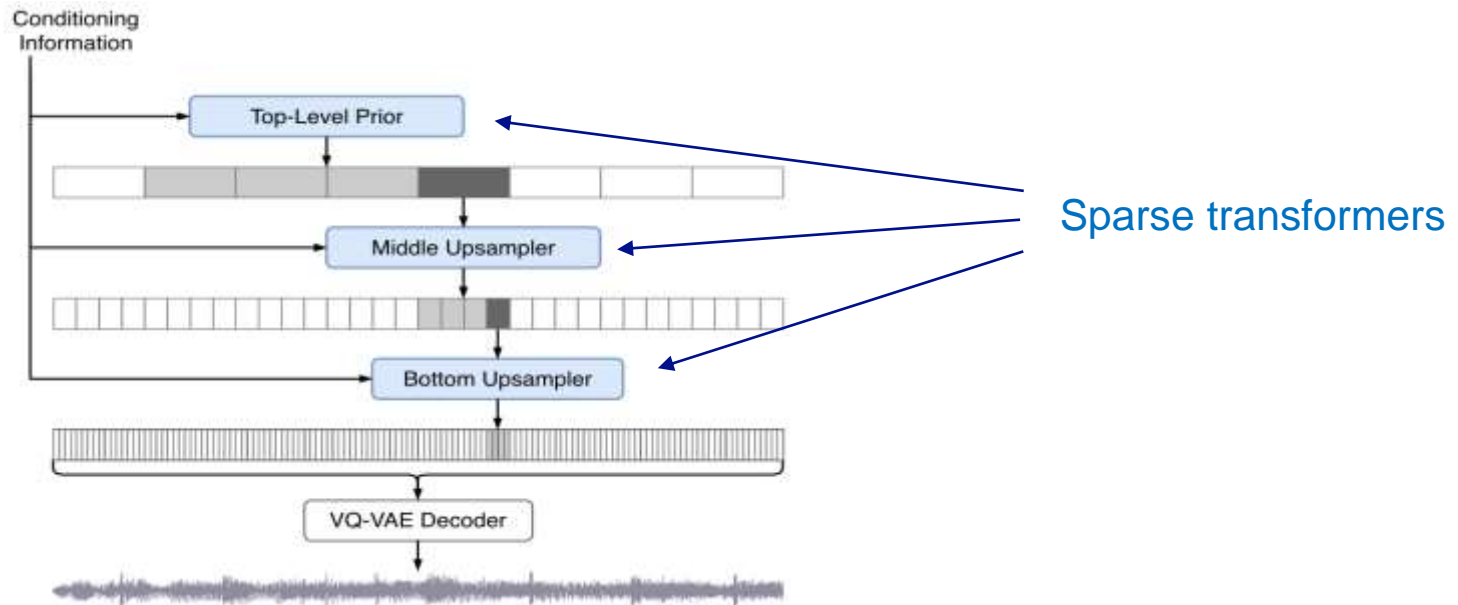
P. Dhariwal & al. "Jukebox: A Generative Model for Music", arXiv:2005.00341

VQ-VAEs in Audio and Music

An example with Jukebox

- Learning the latent prior once the VQ-VAEs are trained

$$\begin{aligned} p(\mathbf{z}) &= p(\mathbf{z}^{\text{top}}, \mathbf{z}^{\text{middle}}, \mathbf{z}^{\text{bottom}}) \\ &= p(\mathbf{z}^{\text{top}})p(\mathbf{z}^{\text{middle}}|\mathbf{z}^{\text{top}})p(\mathbf{z}^{\text{bottom}}|\mathbf{z}^{\text{middle}}, \mathbf{z}^{\text{top}}) \end{aligned}$$



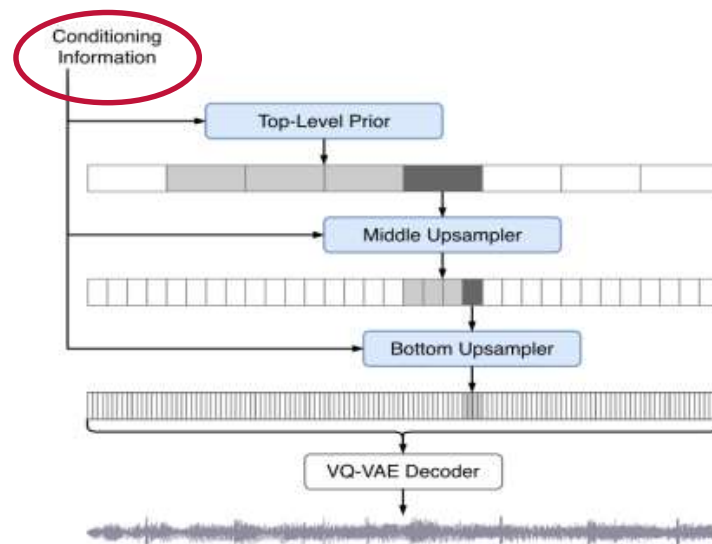
P. Dhariwal & al. "Jukebox: A Generative Model for Music", arXiv:2005.00341



VQ-VAEs in Audio and Music

An example with Jukebox

- Conditioning for controlling the synthesis



- **Artist, Genre, and Timing Conditioning** (to allow to learn patterns that depend on the structure... such as applause at the end)
- **Lyrics Conditioning** (with necessity to learn lyrics/audio alignment)



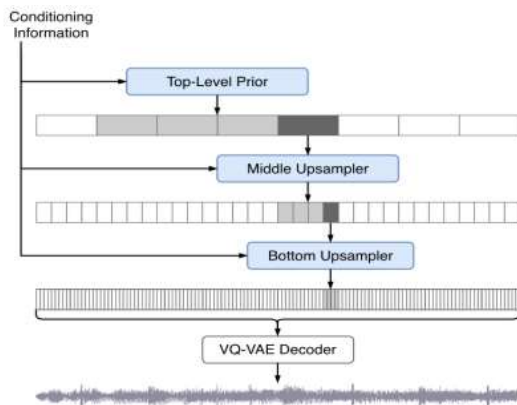
P. Dhariwal & al. "Jukebox: A Generative Model for Music", arXiv:2005.00341



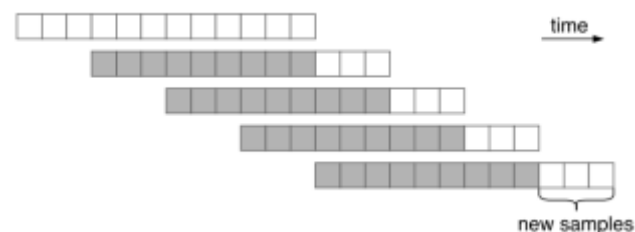
VQ-VAEs in Audio and Music

An example with Jukebox

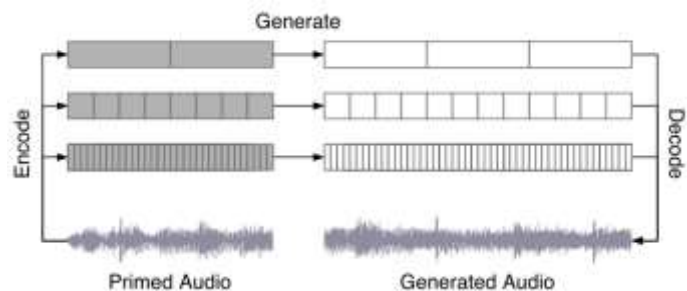
- Sampling methods for generating music



Windowed sampling for modelling sequences longer than initial context



Primed sampling: generate continuations by converting input into the VQ-VAE codes and sampling the subsequent codes in each level.



P. Dhariwal & al. "Jukebox: A Generative Model for Music", arXiv:2005.00341



VQ-VAEs in Audio and Music

An example with Jukebox

- Sound examples
 - Completion (with context of 12s of existing songs in the training)
 - Re-renditions (using pairs of lyrics-artist existing in the training)
 - Generation with novel lyrics (generated by GPT-2)
 - Generation with novel voices (by interpolating existing voice embeddings)
 - Many raw examples at <https://jukebox.openai.com/>
 - Some curated examples at <https://openai.com/blog/jukebox/>
 - One example of continuation with unknown lyrics:
<https://jukebox.openai.com/?song=795460096>
- Original Model is rather slow at sampling (9 hours to render 1' of music)

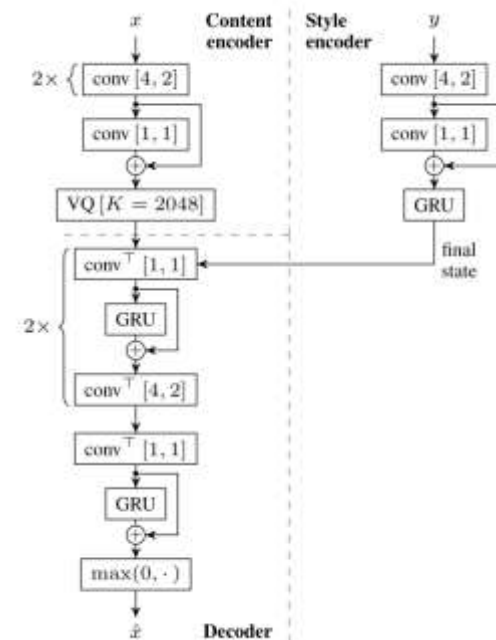
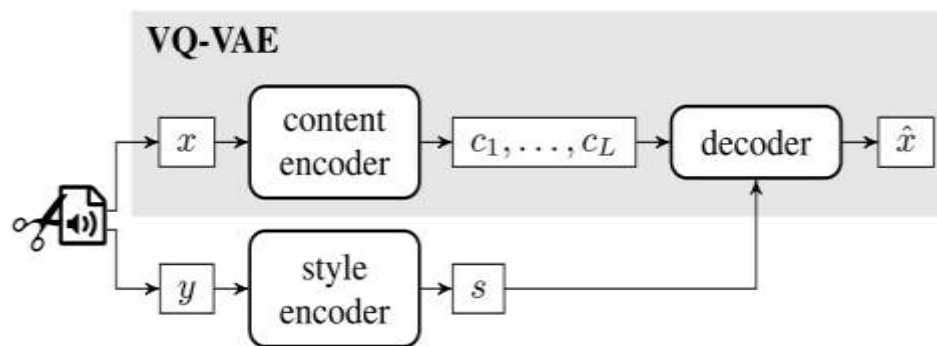


P. Dhariwal & al. "Jukebox: A Generative Model for Music", arXiv:2005.00341

VQ-VAEs in Audio and Music

Another example for one-shot music style transfer

- Content is encoded using a VQ-VAE
- Style is encoded using a self-supervised strategy (y is an *audio-augmented version of a different segment than x , taken from the same recording*)



Ondřej Čířka, Alexey Ozerov, Umut Şimşekli and Gaël Richard. "Self-Supervised VQ-VAE for One-Shot Music Style Transfer." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021.

VQ-VAEs in Audio and Music

Another example for one-shot music style transfer

- Many sound examples at: https://adasp.telecom-paris.fr/rc/demos_companion-pages/cifka-ss-vq-vae/#examples
- Two examples
 1. Synthetic example

Content input



Style input



Target



Output (VQ-VAE)



2. Real example

Content input



Style input



Output (VQ-VAE)

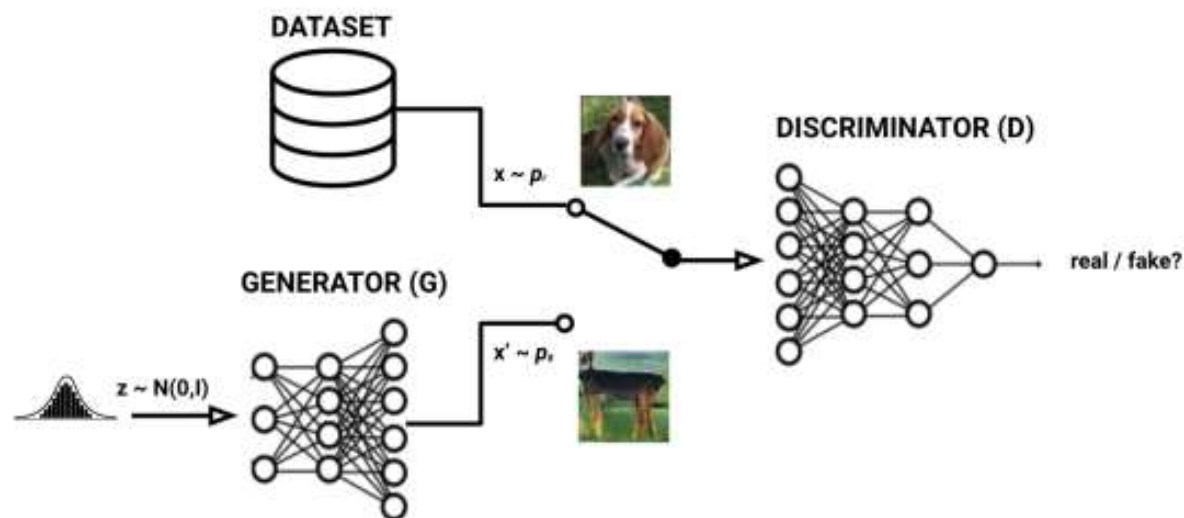


Ondřej Cífka, Alexey Ozerov, Umut Şimşekli and Gaël Richard. "Self-Supervised VQ-VAE for One-Shot Music Style Transfer." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021.



Generative Adversarial Networks (GANs)

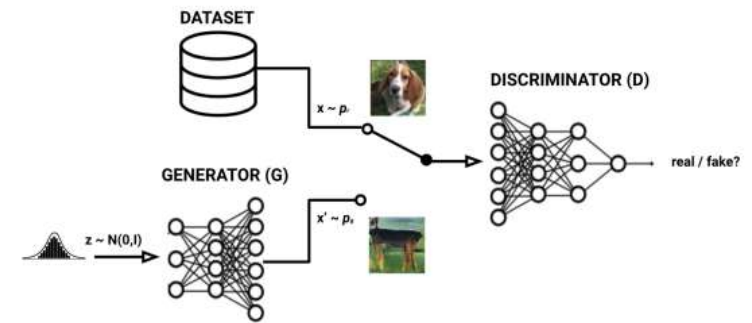
■ Principle of GANs



Ref: Goodfellow, 2014

Figure from J. Nistal, "Exploring generative Adversarial networks for controllable musical audio synthesis, PhD thesis, IP Paris, 2022

Generative Adversarial Networks (GANs)



■ More formally

- a generative network $G_{\theta}(\mathbf{z})$ that outputs $x_g \sim p_g$ from a random input \mathbf{z} . After training, the output should follow the targeted probability distribution p_r
- a discriminative network $D_{\beta}(\mathbf{x})$ trained to predict if the input comes from the real p_r or from the generated distribution p_g
- Optimization problem: a competitive objective

$$\min_{G_{\theta}} \max_{D_{\beta}} V(D_{\beta}, G_{\theta}) = E_{x \sim p_r} [\log D_{\beta}(x)] + E_{x \sim p_g} [1 - \log D_{\beta}(G_{\theta}(z))]$$



Ref: Goodfellow, 2014

Figure from J. Nistal, "Exploring generative Adversarial networks for controllable musical audio synthesis, PhD thesis, IP Paris, 2022"



Generative Adversarial Networks (GANs)

■ Principle of conditional GANs for audio synthesis

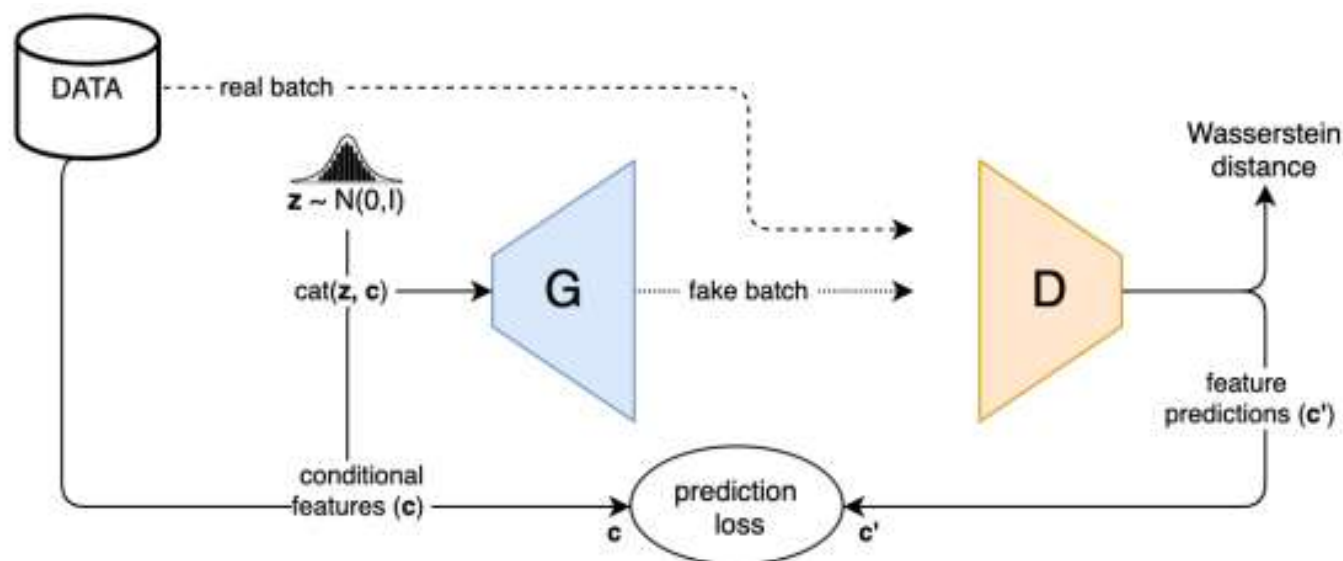
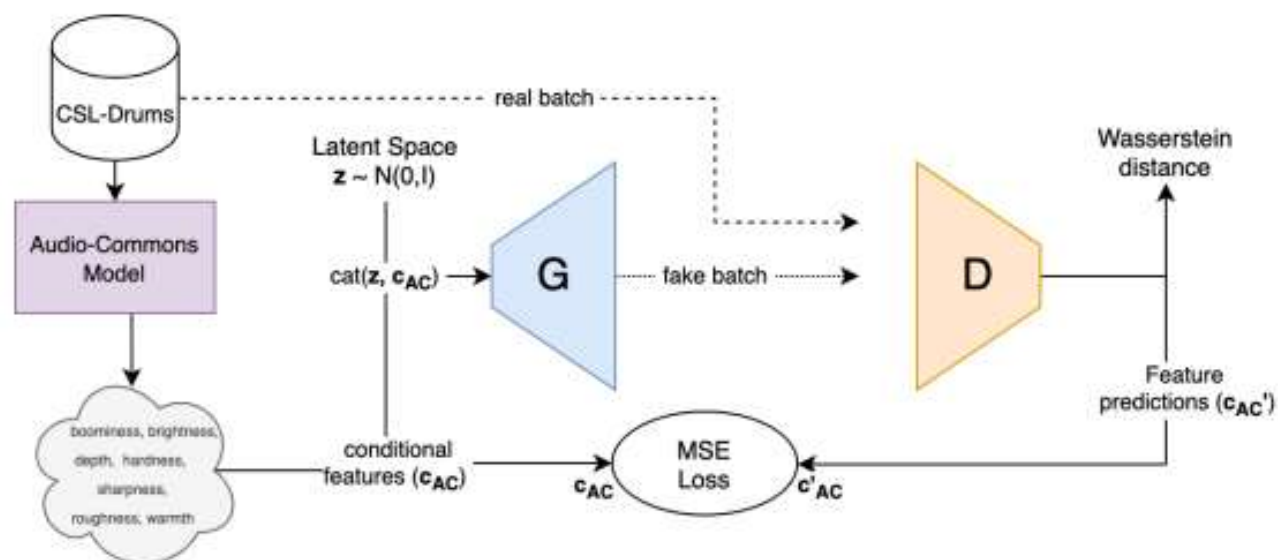


Figure from J. Nistal, "Exploring generative Adversarial networks for controllable musical audio synthesis, PhD thesis, IP Paris, 2022



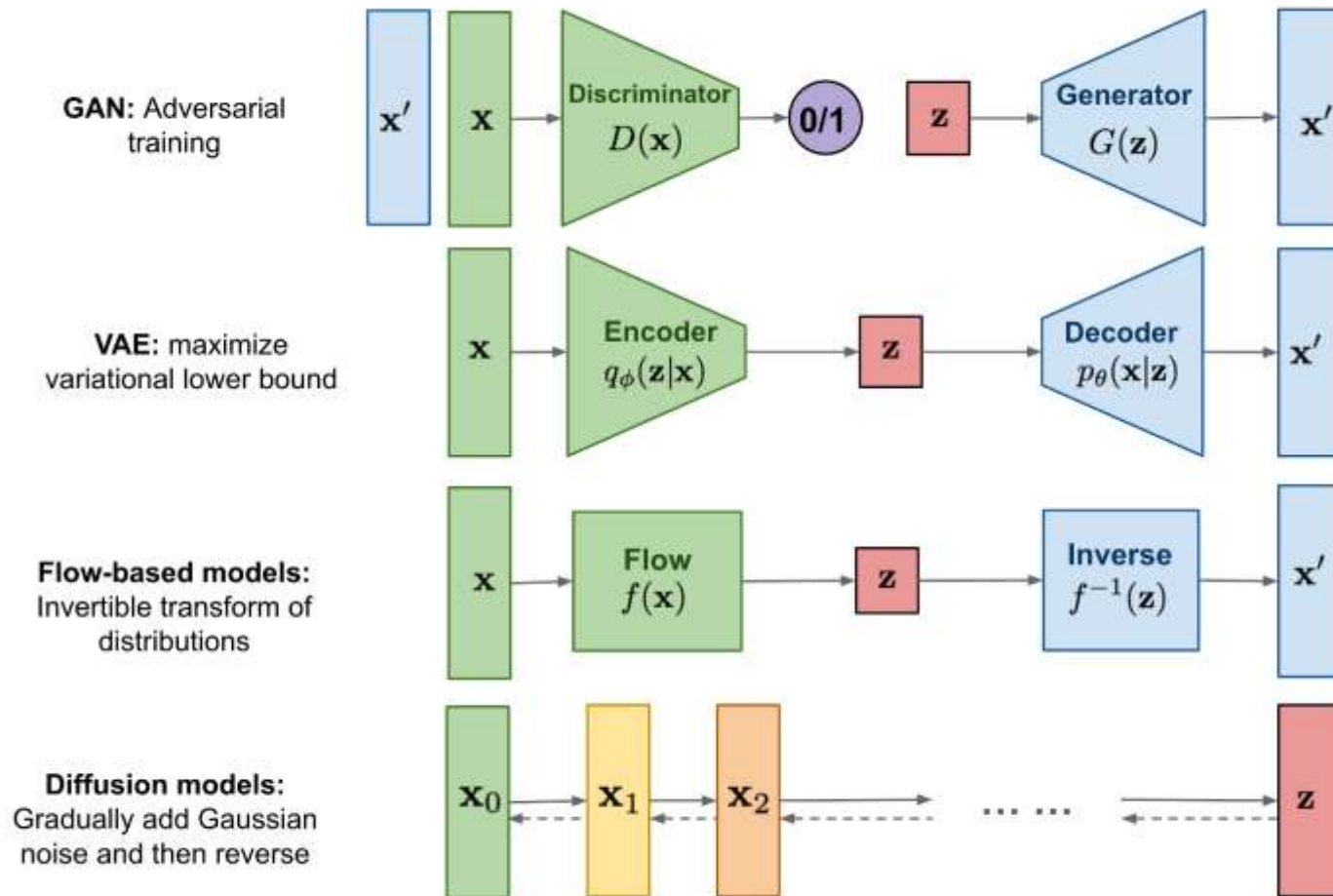
Generative Adversarial Networks (GANs)

- **DrumGAN: Synthesis of Drum sounds with timbral feature Conditioning using GANs synthesis**



Nistal, J., Lattner, S., and Richard, G., "DrumGAN: Synthesis of Drum Sounds with Perceptual Feature Conditioning using GANs," in Proceedings of the 28th International Society for Music Information Retrieval, ISMIR, 2020.

A large variety of generative models ...



L. Weng, What are diffusion models ?, 2021. <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>



A few examples of flow-based audio synthesis models

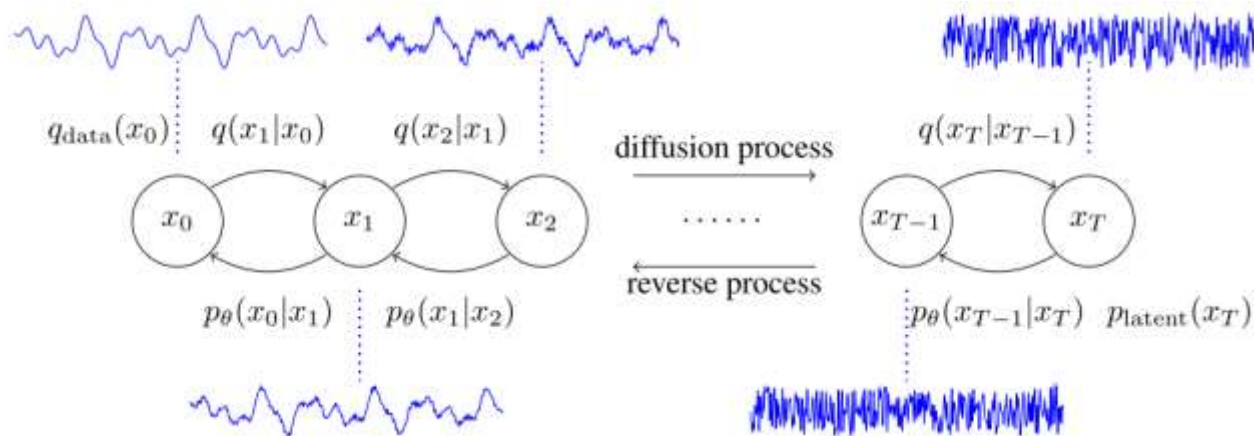
Arch.	Name	Audio representation	Data	Conditioning	
NFs	Parallel waveNet	van den Oord et al., 2018a	waveform	speech	text pitch
	ClariNet	Ping et al., 2018	waveform	speech	text
	FlowwaveNet	Kim et al., 2018	waveform	speech	text Mel spec.
	waveGlow	Prenger et al., 2018	waveform	speech	text Mel spec.
	waveFlow	Ping et al., 2020	waveform	speech	text Mel spec.
	Blow	Serrà et al., 2019	waveform	speech	speaker ID



J. Nistal, PhD thesis, 2022



Diffusion models for audio synthesis ...



- Based on two processes: the diffusion process, and the reverse process

- The **diffusion process** is defined by a fixed Markov chain from data x_0 to the latent variable x_T

$$q(x_1, \dots, x_T | x_0) = \prod_{t=1}^T q(x_t | x_{t-1})$$

where each of $q(x_t | x_{t-1})$ is fixed to $\mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I)$ for a small positive constant β_t

- The **reverse process** gradually converts the white noise signal into audio waveform through a Markov chain:

$$p_{\text{latent}}(x_T) = \mathcal{N}(0, I), \text{ and } p_{\theta}(x_0, \dots, x_{T-1} | x_T) = \prod_{t=1}^T p_{\theta}(x_{t-1} | x_t),$$



Kong, Z., Ping, W., Huang, J., Zhao, K., & Catanzaro, B. (2020). DiffWave: A Versatile Diffusion Model for Audio Synthesis. ArXiv, abs/2009.09761.

Diffusion models for audio synthesis ...

■ The models are often strongly conditioned

- Example: wavgrad, specgrad conditioned on mel-spectrogram

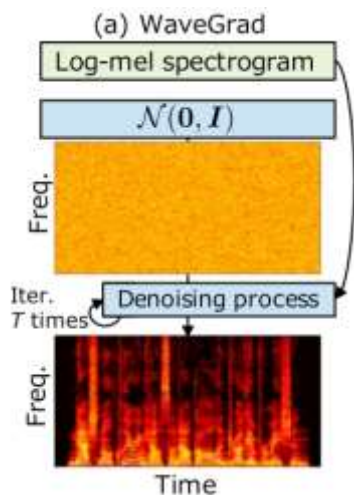


Illustration of the diffusion process (50 iterations)



Sound examples

reference



wavgrad



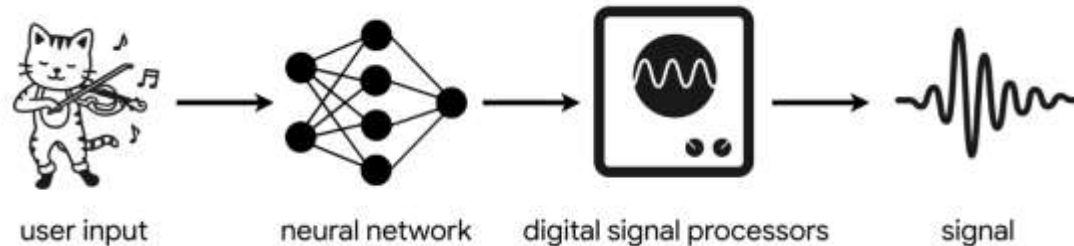
N. Chen & al. "WaveGrad: Estimating gradients for waveform generation," in Proc. ICLR, 2021.

Koizumi, Yuma et al. "SpecGrad: Diffusion Probabilistic Model based Neural Vocoder with Adaptive Noise Spectral Shaping." Interspeech (2022).

Towards Hybrid deep learning approaches

■ Coupling model-based and deep learning

- For example, using deep learning for learning the parameters of a signal processing model

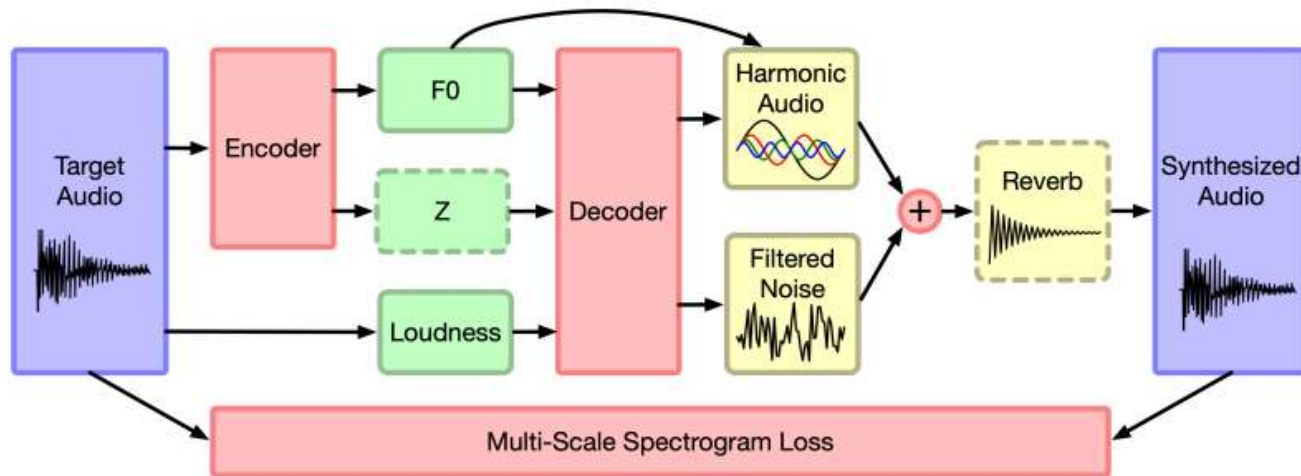


J. Engel & al., "DDSP: Differentiable Digital Signal Processing," in Int. Conf. on Learning Representations (ICLR), 2020.



Towards Hybrid deep learning approaches

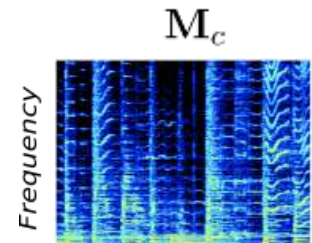
- The example of DDSP



- A multi-scale spectral loss $\mathcal{L}_{rec} = \sum_c \mathcal{L}_c$

$$\text{With } \mathcal{L}_c = \|\mathbf{M}_c - \tilde{\mathbf{M}}_c\|_1 + \|\log(\mathbf{M}_c) - \log(\tilde{\mathbf{M}}_c)\|_1$$

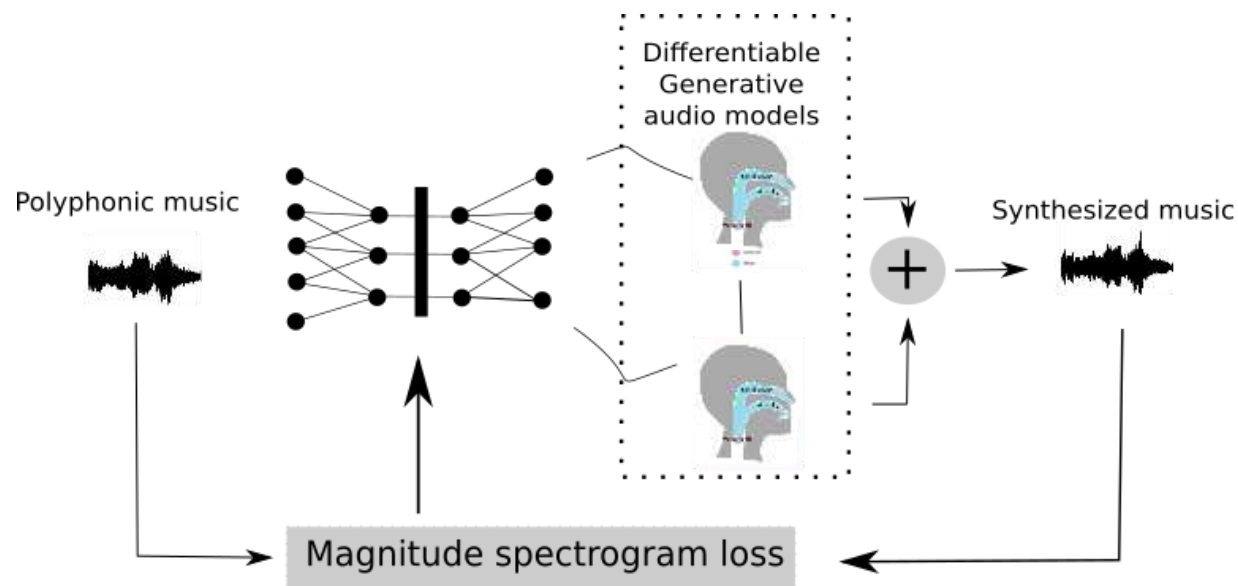
and with $c = [2048, 1024, 512, 256, 128, 64]$ indicates the FFT size used to compute the STFT.



J. Engel & al., "DDSP: Differentiable Digital Signal Processing," in Int. Conf. on Learning Representations (ICLR), 2020.

Towards Hybrid deep learning approaches: DDSP extensions and others...

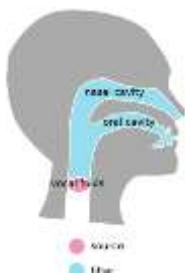
- An example for unsupervised singing voice separation



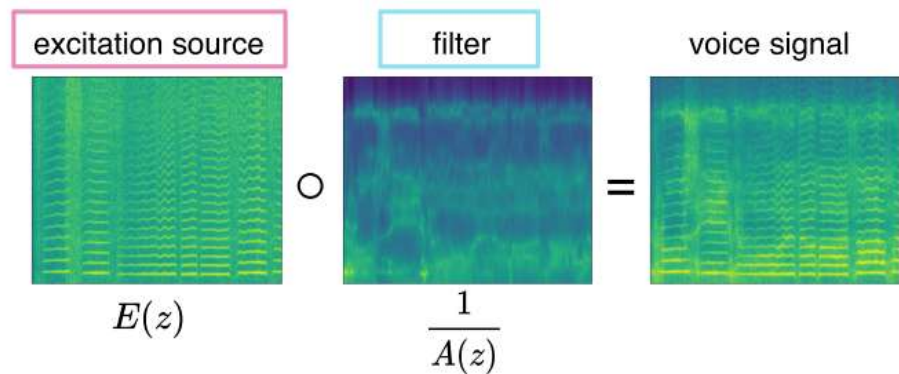
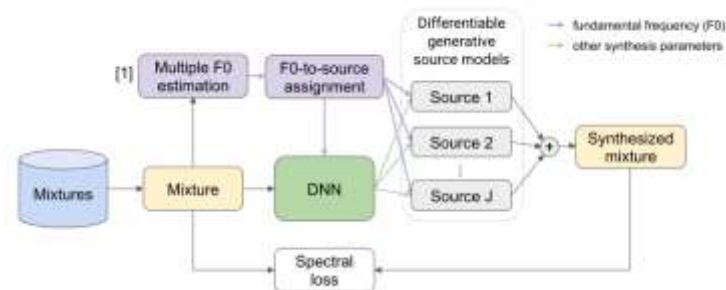
K Schulze-Forster, G. Richard, L. Kelley, C. Doire, R Badeau Unsupervised Music Source Separation Using Differentiable Parametric Source Models, IEEE Trans. On AASP, 2023

Parametric source models

Singing voice as a source / filter model :



- source = vibration of vocal folds
- Filter = resonances of vocal/nasal



J. Engel, C. Gu, A. Roberts et al., "DDSP: Differentiable digital signal processing," in Proc. Int. Conf. Learning Representations, 2019



A short audio demo

■ A short demo at

- <https://schufo.github.io/umss/>
- Ou [lien local](#)



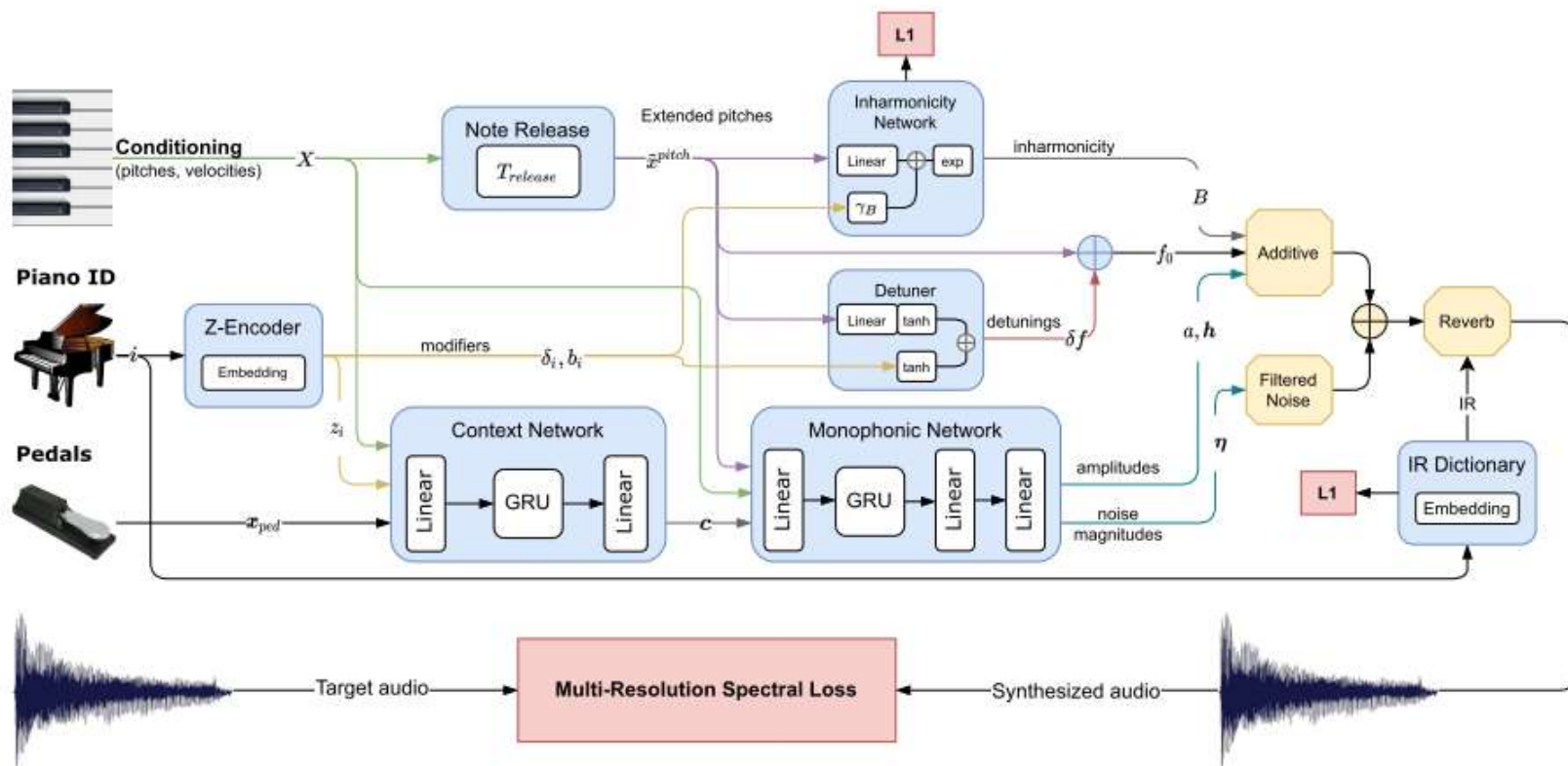
■ Beyond source separation:

- **The synthesis model allows for sound transformation**



Towards Hybrid deep learning approaches: DDSP extensions and others...

- A differentiable piano model for neural synthesis

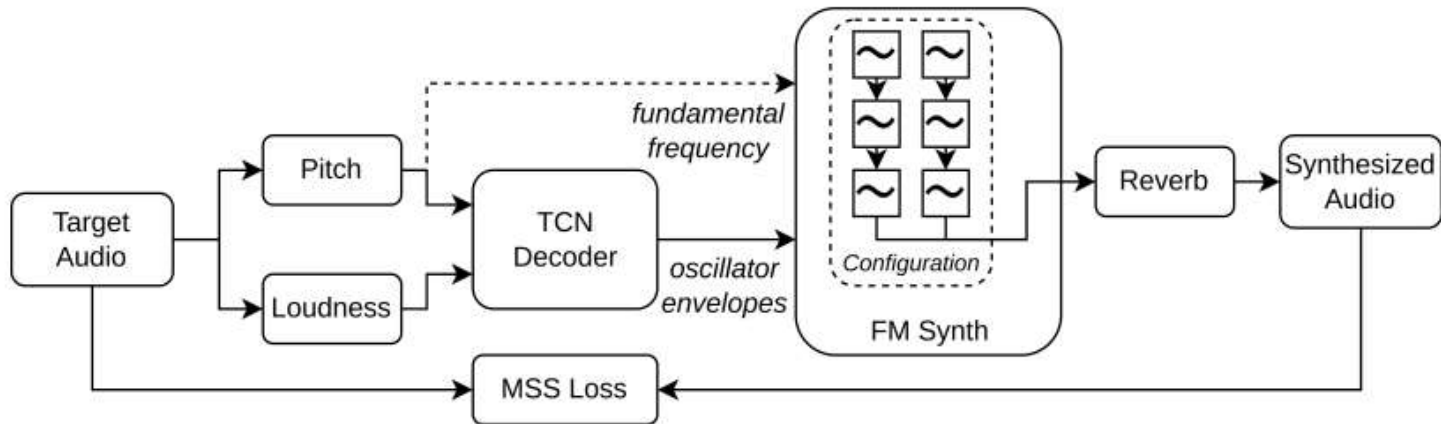


Lenny Renault, Rémi Mignot, Axel Roebel. Differentiable Piano Model for MIDI-to-Audio Performance Synthesis. 25th International Conference on Digital Audio Effects (DAFx20in22), Sep 2022, Vienna,



Towards Hybrid deep learning approaches: DDSP extensions and others...

- DDX7: A differentiable FM synthesizer
(fixed frequency ratios and max range modulation index)



Caspe, F.S., Mcpherson, A.P., & Sandler, M.B. (2022). DDX7: Differentiable FM Synthesis of Musical Instrument Sounds. ArXiv, abs/2208.06169.



Evaluation of audio synthesis

■ Some of the important questions:

- What should we evaluate (Quality/naturalness ? Diversity of sounds, .. Artistic value of sound ?)
- How do we evaluate such dimensions ?



Evaluation of audio synthesis

■ Objective evaluation

- Evaluation metrics typically rely on:
 - either mathematical formulations of a success measure,
 - or (especially with GANs) using a separate neural network to identify if the model is working appropriately.

■ Four main objective characteristics evaluated

- Reconstruction metrics,
- Sample diversity measures,
- Distribution distance measures,
- Measures derived from subjective evaluation methods.

Vinay, Ashvala & Lerch, Alexander. (2022). Evaluating generative audio systems and their metrics. 10.48550/arXiv.2209.00130.



Evaluation of audio synthesis

Objective metrics

■ Reconstruction metrics

- ℓ_1 or ℓ_2 metrics, multiscale distances on spectrogram
- ...but not well aligned with perception

■ Sample diversity metrics (to measure the quality of the generator):

- **Number of statistically Different Bins (*NDB/k*)**
 - Based on a clustering algorithm (k-means of the training samples)
 - Test samples are assigned to the cluster using ℓ_2 distance
 - A two-sample t-test on each bin identifies the statistically different bins.
 - The final NDB score is given by counting the number of statistically different bins and dividing by the number of clusters.



Vinay, Ashvala & Lerch, Alexander. (2022). Evaluating generative audio systems and their metrics. 10.48550/arXiv.2209.00130.

E. Richardson and Y. Weiss, "On GANs and GMMs," in Advances in Neural Information Processing Systems, Available: <https://proceedings.neurips.cc/paper/2018/file/0172d289da48c48de8c5ebf3de9f7ee1-Paper.pdf>



Evaluation of audio synthesis

Objective metrics

- **Sample diversity metrics** (to measure the quality of the generator):
 - **Inception scores (IS):**
 - Use a classifier to automatically evaluate whether :
 - the output of a synthesizer is of reasonable quality (the generated sound can be easily classified)
 - the synthesizer captures the diversity of samples in the dataset (the synthesizer can output sounds that easily identifiable for many different classes/labels)
 - Basically evaluate if the difference between the probability distribution of predicted labels for the generated sound and the marginal distribution of the labels from the generated data is large



Evaluation of audio synthesis

Objective metrics

- **Distribution distance metrics:** (to measure if the distribution of the generated data is close to those of the real data):

Kernel Inception Distance

- Similar to inception distances but computed on the embeddings (for neural audio synthesis) – usually of last layer.
- Distance computed using squared Maximum Mean Discrepancy between representations of the last layer of the same Inception model

Frechet Audio Distance

- Compares the statistics of real and generated data computed from an embedding layer of a pre-trained VGG-like model

$$FAD = \|\mu_r - \mu_g\|^2 + \text{tr}(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g})$$

K. Kilgour & al. "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in Interspeech 2019. ISCA, pp. 2350–2354.



Evaluation of audio synthesis

Subjective metrics

■ Subjective evaluation :

- Based on human ratings of a pre-selected number of generated examples.
- Popular measure: Mean Opinion Score (MOS)
 - Initially use to measure quality of transmission [1]
 - Based on Likert scale in the range [1 – 5] (that is with « category »)

Rating	Quality	Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible, but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying, but not objectionable
1	Bad	Very annoying and objectionable

- Participants are asked to rate sounds on for example their “sound quality, ease of intelligibility, and speaker diversity” where 1 indicates bad and 5 indicates excellent.
- Provides an absolute rating of quality (no direct comparison with references, no direct comparison between methods)

[1] “Methods for subjective determination of transmission quality,” ITU-T Recommendation P.800, Aug. 1996.



Evaluation of audio synthesis

Subjective metrics

■ Subjective evaluation :

- MUSRHA (*MU*ltiple *S*timuli, *H*idden *R*eference and *A*ncor) tests to compare models.

Example of a typical interface for the test (from Audio Labs, Fraunhofer)

The screenshot shows the webMUSHRA interface for a Mono Trial. At the top, there is a 'Stop' button and a progress bar with time markers 1:28 and 3:40. Below the progress bar is a waveform with a highlighted orange segment. Underneath the waveform are five conditions: Reference, Cond.1 C1, Cond.2 C2, Cond.3 C3, Cond.4 reference, and Cond.5 anchor35. Each condition has a 'Play' button. Below the conditions is a vertical bar chart with five bars representing the quality scores for each condition. The y-axis is labeled 'Excellent', 'Good', 'Fair', 'Poor', and 'Bad'. The x-axis shows scores: 70, 60, 60, 100, and 60. A 'Next' button is at the bottom. Logos for AUDIO LABS, Fraunhofer, and FAU are visible at the bottom right.

ITU-R BS.1116 "Methods for the subjective assessment of small impairments in audio systems".



Evaluation of audio synthesis

■ Some remarks

- Several objective metrics but do not correlate well with real perception
- Subjective metrics are good but costly to obtain and the perceptual tests need to be carefully designed (choice of samples, selection of participants, ...)



Conclusion

- Audio synthesis is an attractive field (many recent work on all aspects)
- Audio synthesis benefits from the advances in image synthesis

■ Some other aspects not discussed in this lecture

- Audio Style transfer (Groove2groove,...)
- Music generation from image (soundof pixel,...)
- Music generation from Text (MusicLM, AudioLM,...)

Ondrej Cifka, Umut Simsekli, Gaël Richard, "Groove2Groove: One-Shot Music Style Transfer with Supervision from Synthetic Data", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2638-2650, 2020

Zhao, Hang et al. "The Sound of Pixels." *ArXiv abs/1804.03160* (2018)

Agostinelli, Andrea et al. "MusicLM: Generating Music From Text." *ArXiv abs/2301.11325* (2023)

Liu, Haohe et al. "AudioLDM: Text-to-Audio Generation with Latent Diffusion Models." *ArXiv abs/2301.12503* (2023)

