A messy intro to Neural Audio Synthesis

exploring practices and quirks

Giacomo Lepri *(he/him)*

Musician
Composer
Instrument Designer
Researcher

Specialised in not being specialized

NIME

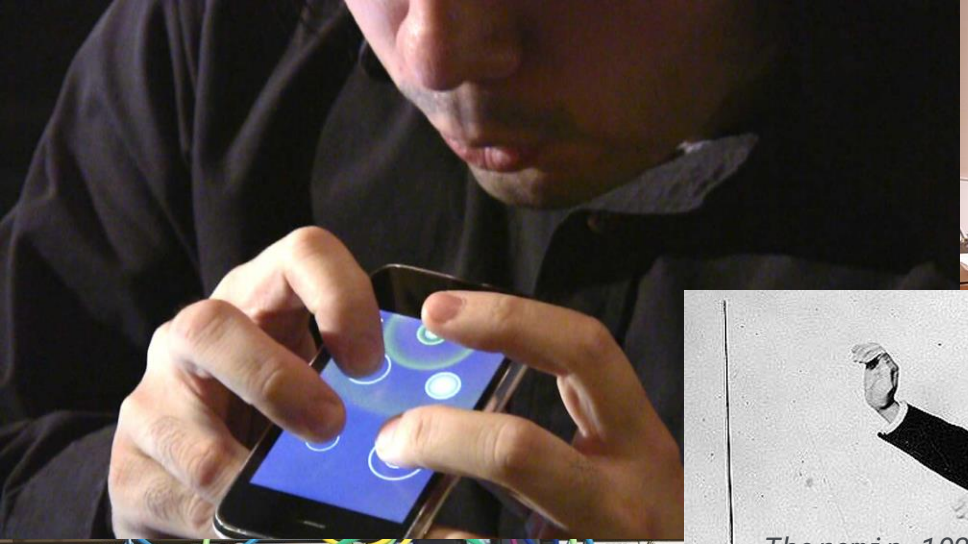
New Interfaces for Musical Expression

Halldorophone

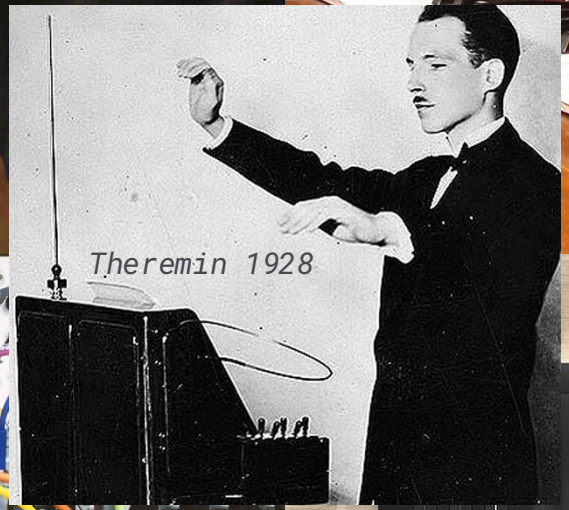

Performing with Computers
Pamela Z

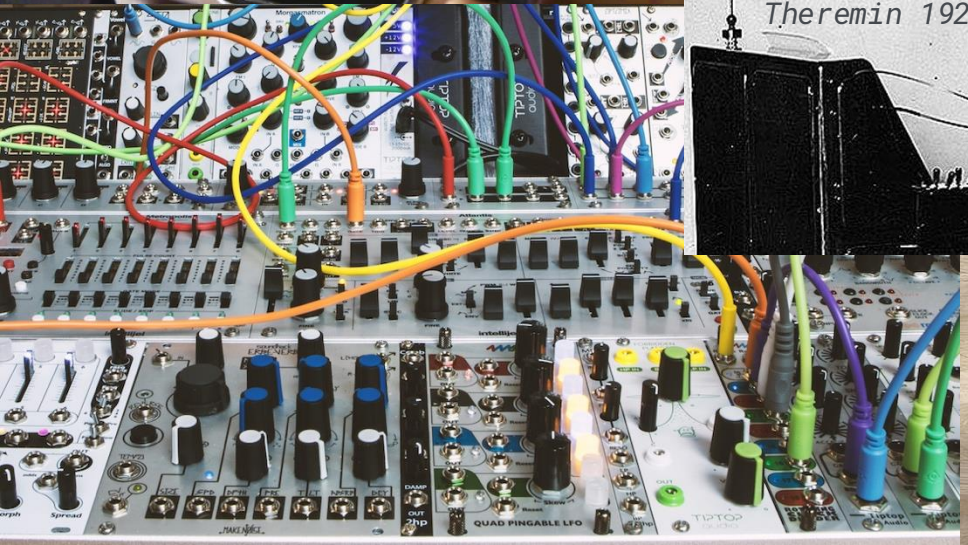

How to play
a computer?


PLOrk

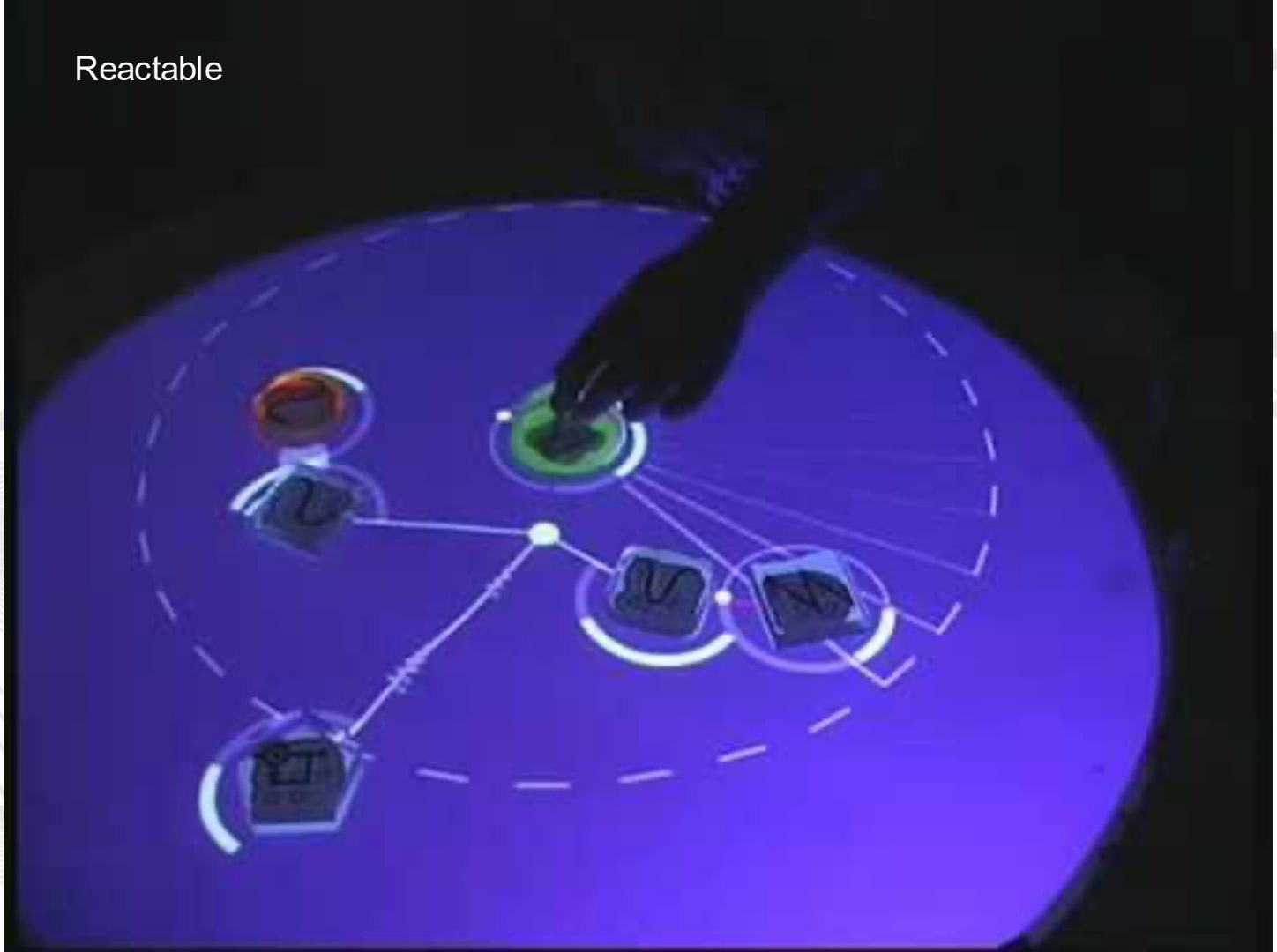



OSCILLATOR BANK
MIXER
MODIFIERS
OUTPUT
R. A. MOOG


Theremin 1928


QUAD PINGABLE LFO

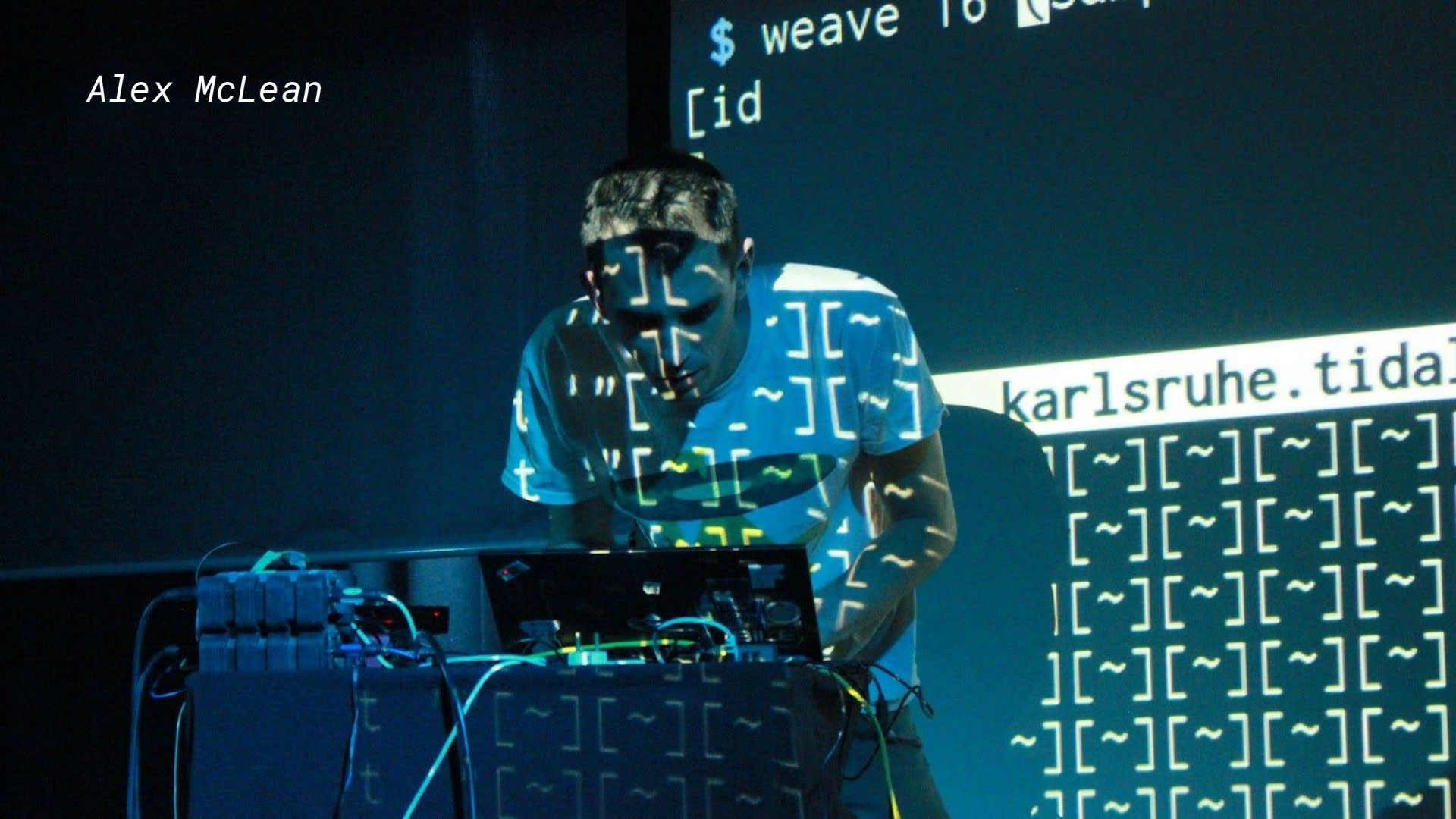
Reactable




Extended keyboard techniques:
Crescendos from silence
(gradual key press)

Halldorophone
by Halldór Úlfarsson


Enrique Tomás – Tangible Scores


Alex McLean
$ weave 16
[id
karlsruhe.tida

Pompom Musical Instrument
Sam Topley

WHY!?

Music creation
Cultural studies
Music accessibility
Human cognition
Technology research

Music and research

Research **on**	Studying musical phenomena (descriptive/analytical)
Research **for**	Creating knowledge/tools to support or improve musical practice
Research **through**	Generating knowledge by making and reflecting on praxis Studying broader phenomena through musical practice and interaction

What Counts as ‘Creative’ Work? Articulating Four Epistemic Positions in Creativity-Oriented HCI Research

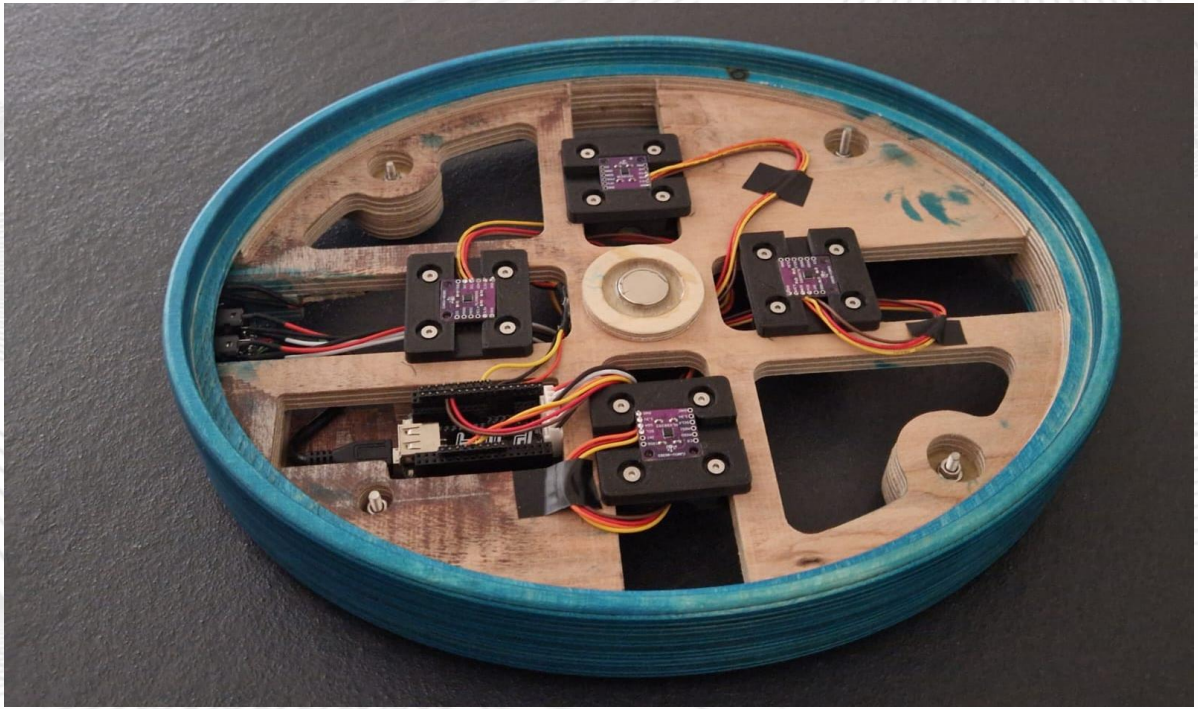
Stacy Hsueh, Marianela Ciolf Felice, Sarah Fdili Alaoui, and Wendy E. Mackay

CHI Conference on Human Factors in Computing Systems (CHI ’24).

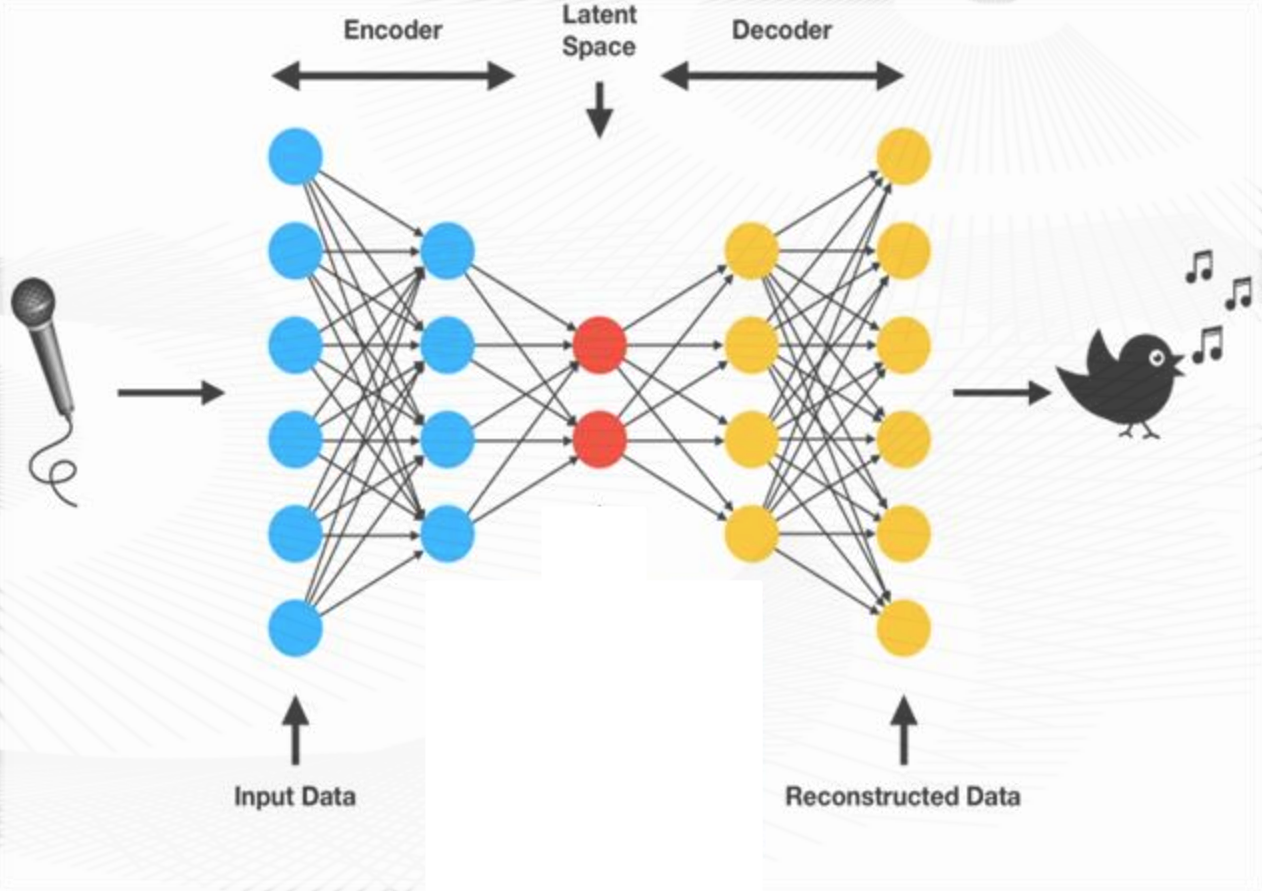
Creative work as	Characteristics	Unit of analysis	Site
Problem-solving	To develop heuristics to navigate solution space	Systems models	Structured activities: e.g. engineering, architecture
Cognitive emergence	To generate new ideas, make associations, combine concepts	Phases of a process	Ideation activities
Embodied action	To engage with embodied knowledge and the dynamic material world	Relations between body and world	“Alt” sites: e.g. everyday resourcefulness, craftwork
Tool-mediated expert activity	To perform creative tasks as mediated by tools	Common tasks	Creative practices in which one can develop expertise: e.g. graphic design

Hardware

Wooden board with four magnetic attractors
XYZ magnetic field detection on each attractor
Bela Board for embedded synthesis and/or OSC data forwarding



Neural Audio Synthesis (NAS)

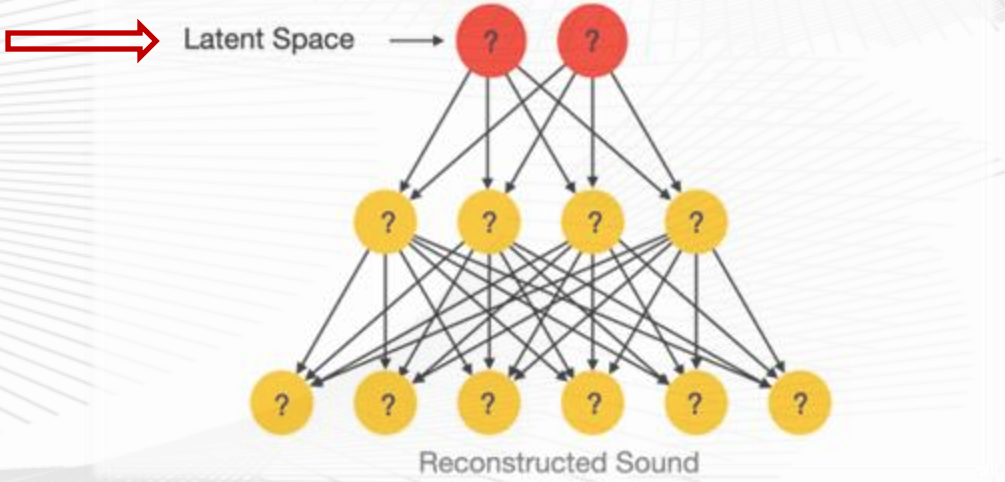


NAS musical affordances

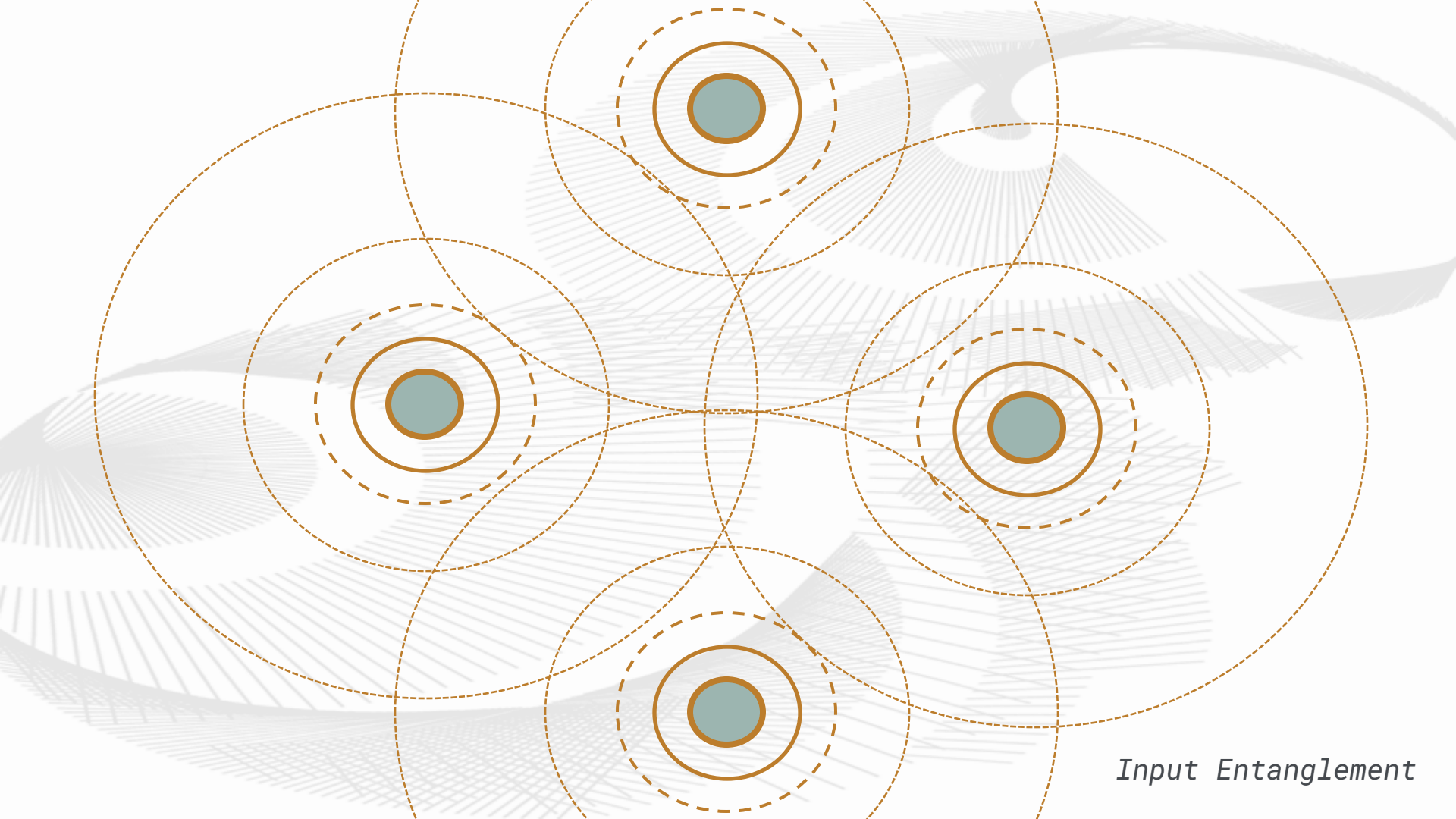
***Continuity**: latent spaces are active at all times, we cannot turn them on or off with a trigger.*

***Arbitrariness**: sound parameters (frequency, amplitude etc…) are arbitrarily distributed by the model during training and explored by the performer/composer afterwards.*

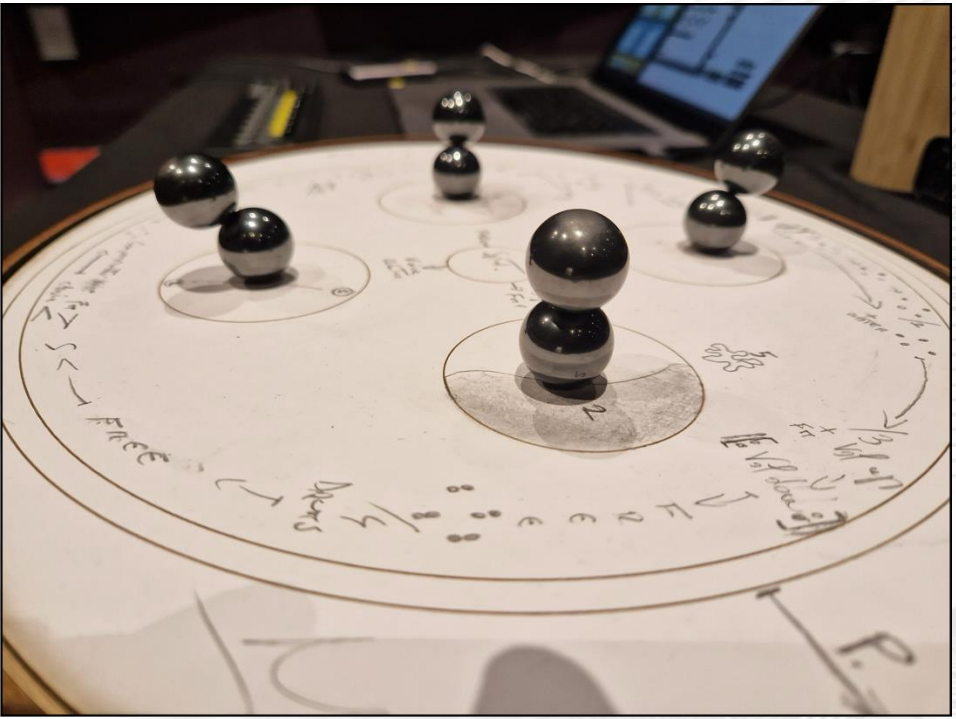
***Entanglement**: sound parameters are intertwined, each latent dimension controls a combination of parameters.*

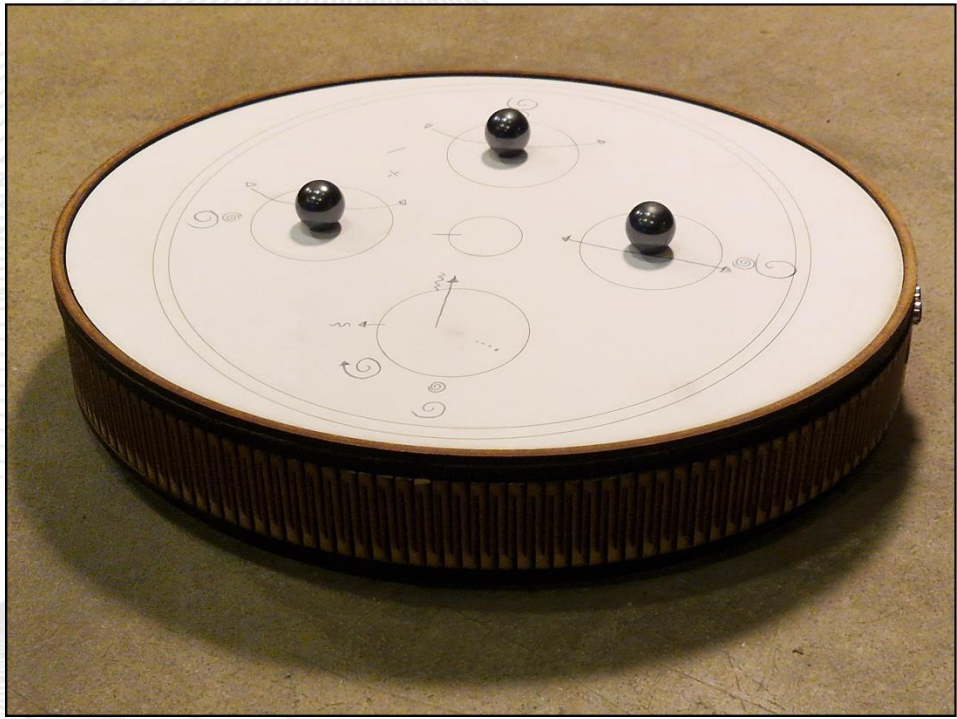


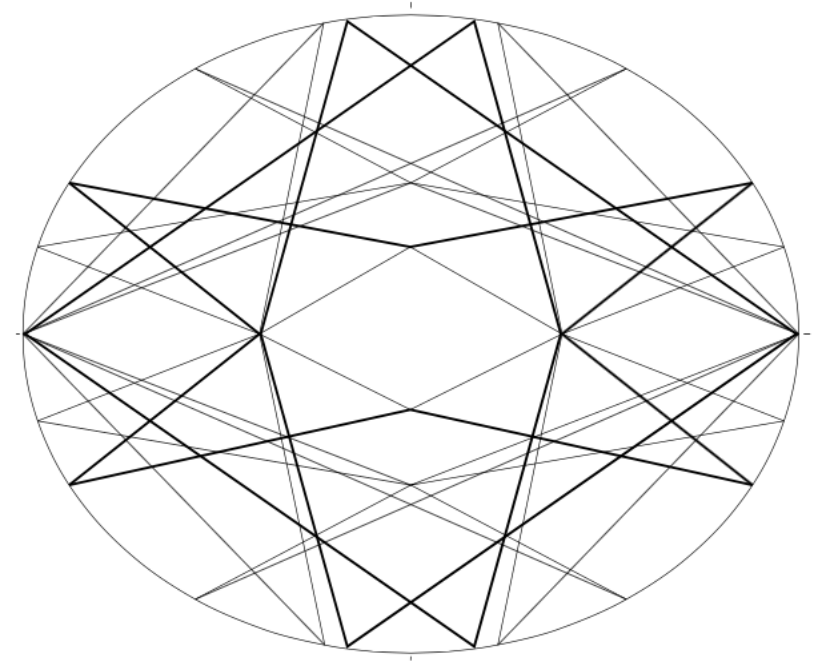
Latent spaces are arbitrarily populated, entangled and continuous

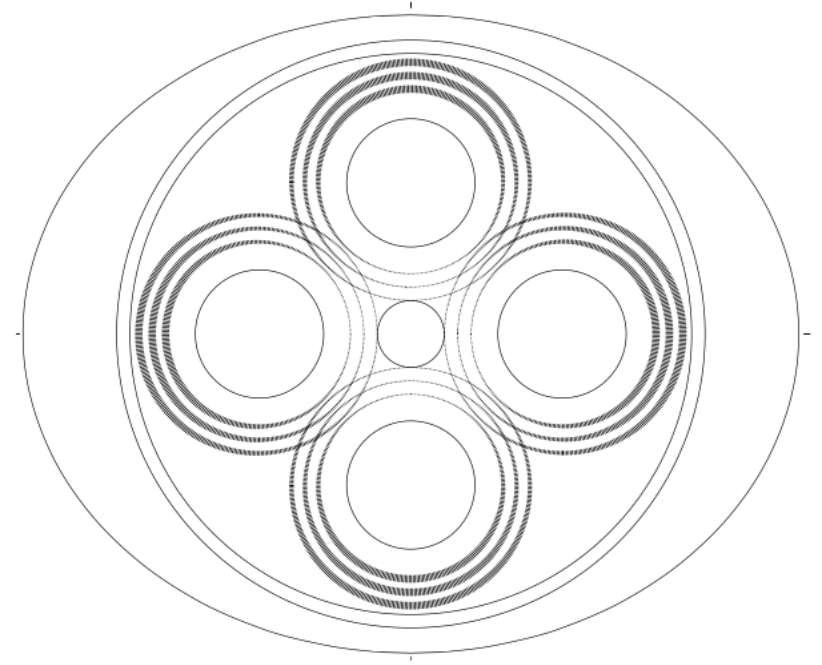

Input Entanglement

Embodied Sketching

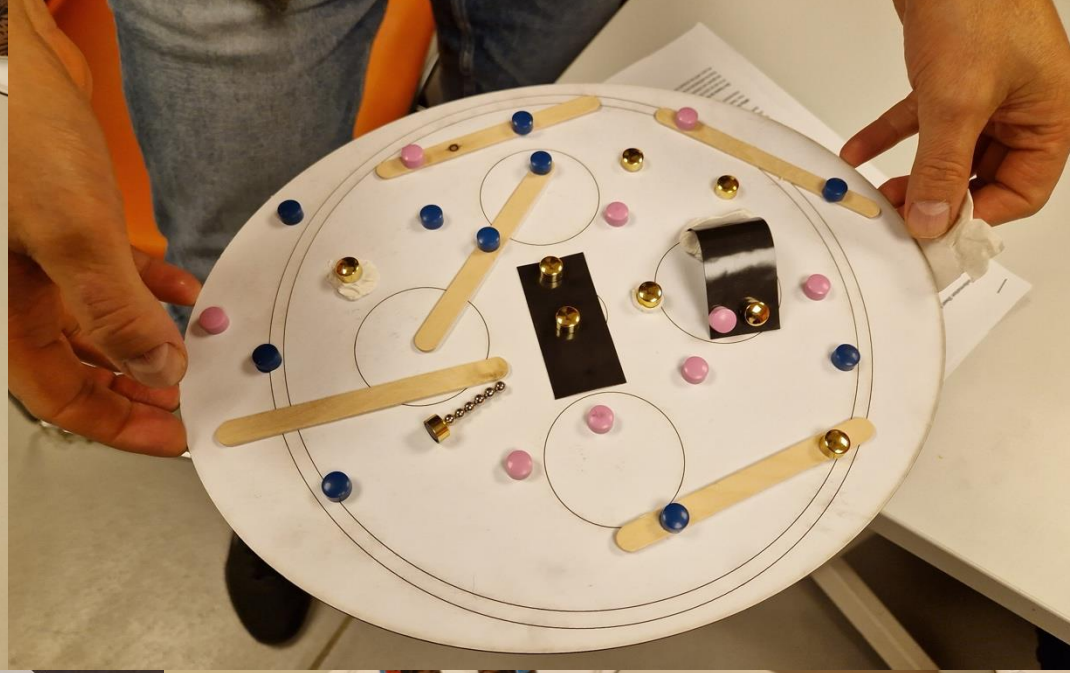


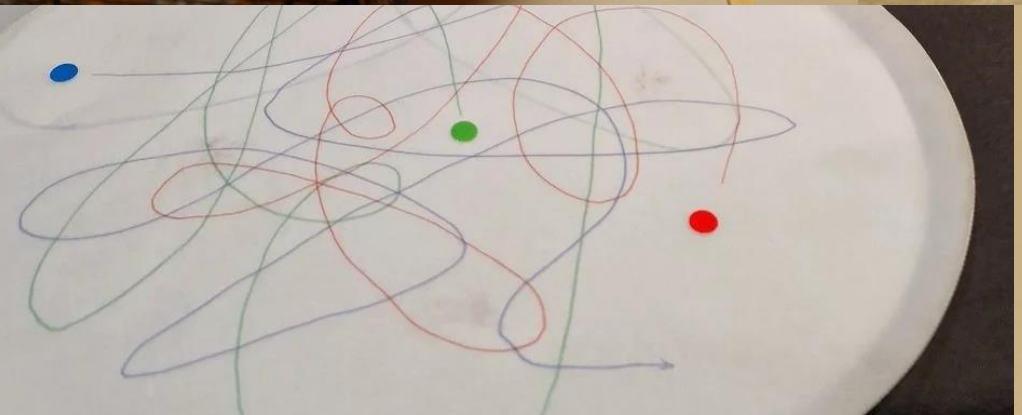







DO&PSY
CHALK





What is RAVE:

RAVE is a Realtime Audio Variational autoEncoder optimized for fast, high-quality audio synthesis

Key Features:

- *Real-time synthesis at 48kHz*
- *Compact latent representations for efficient manipulation (Max/MSP and Pure Data)*
- *Combines autoencoding and adversarial techniques for high-quality output*
- *Uses multiband decomposition to process and reconstruct audio efficiently*
- *Latency management: ensures smooth real-time audio synthesis.*

Two-Stage Training Process

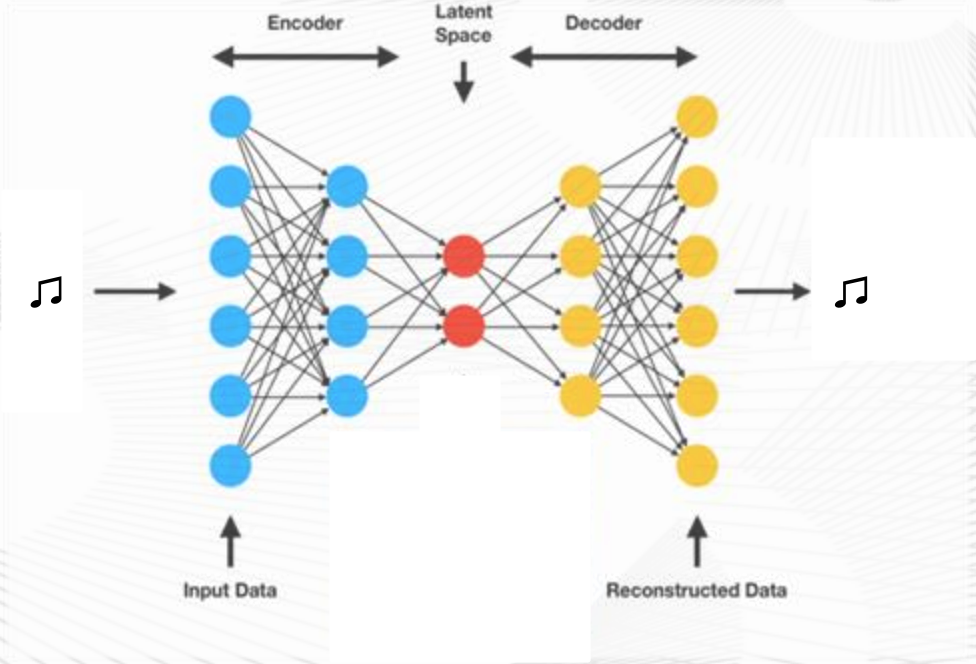
Representation Learning: focus on learning high-level audio attributes

- Autoencoder

Adversarial Fine-Tuning: ensures natural and high-quality audio reconstruction using GANs.

- Discriminator

What is an Autoencoder?



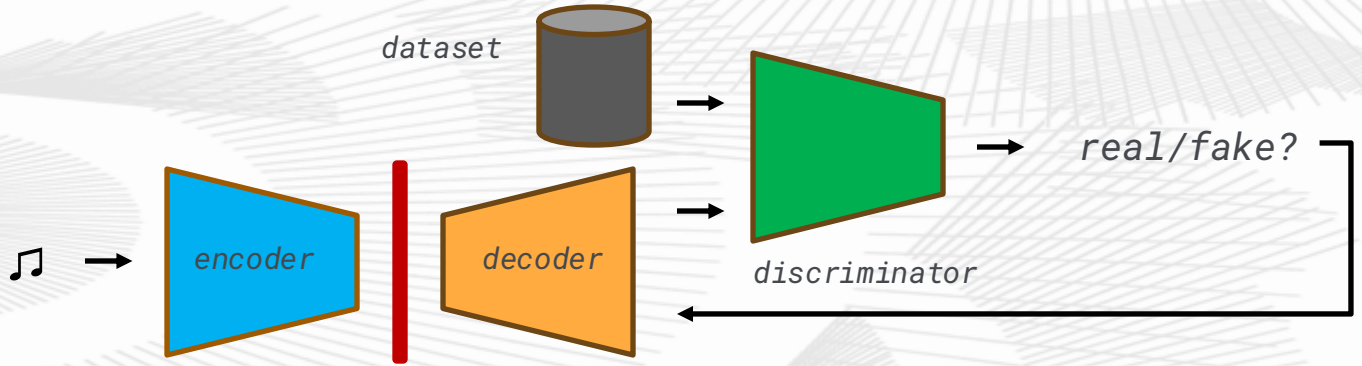
Reduces audio into a compact 128-dimensional latent space (by default) e.g. [0 0.6 0.8]

Reconstructs audio from latent representation using upsampling.

Adversarial Training in RAVE

RAVE incorporates a GAN-like discriminator to improve audio synthesis

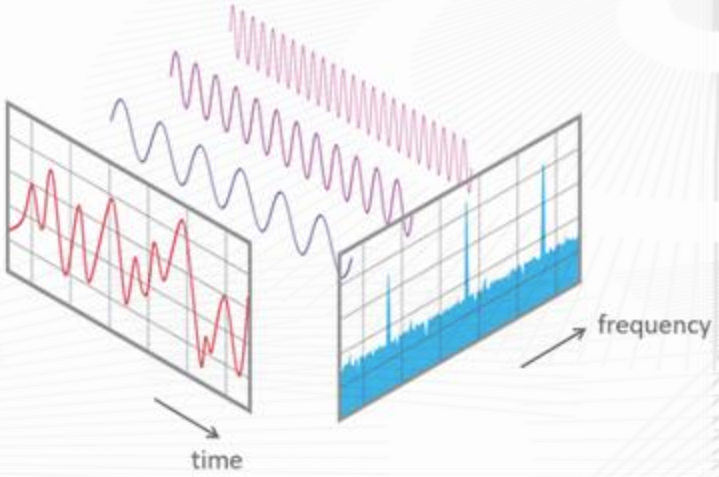
Adversarial Fine-Tuning:
Uses a discriminator to distinguish real vs. synthesized audio.



Goal is to increase realness
Produces natural and artifact-free audio

- *Enhances audio realism by teaching the model to fool the discriminator.*
- *Combined with Autoencoding, Adversarial methods complement the variational autoencoder to refine reconstruction fidelity.*

Frequency Band Decomposition



Why Frequency Bands?

For efficient processing and reconstruction (each band is downsampled for reduced computational load).

Mimics how humans perceive sound (e.g., low and high frequencies are processed differently).

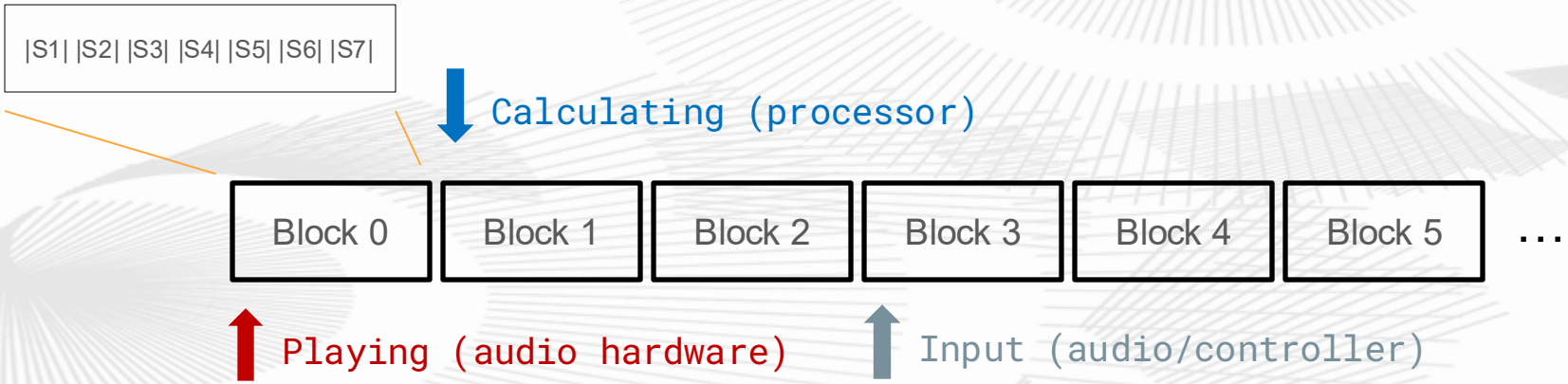
How RAVE Uses Bands:

PRE-PROCESSING STEP: audio encoded in frequency bands and processed in latent space and reconstructed.

Advantage: Improves synthesis quality and performance, allows 48kHz synthesis.

Latency challenge

Real-time synthesis demands processing at or below playback time.



2048 samples each block = 43ms

Efficient buffering ensures continuous, glitch-free synthesis.

Max and PD

Some of the dimension are washed out via PCA principal component analysis – choose the most important dimensions (dimensionality reduction)

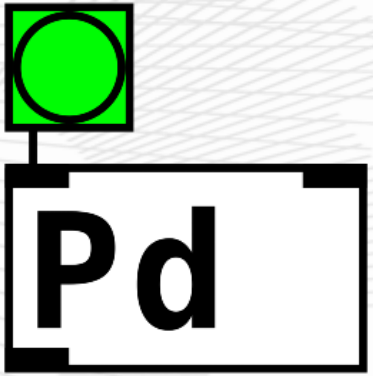
- Separation into informative and non-informative parts
- Allows manipulation of only the informative parts.

Download and install Pd

- puredata.info/downloads/pure-data

Here you have a good start guide

- https://puredata.info/docs/StartHere



This patch generates a sine tone and controls its amplitude over time using a simple envelope.

The slider sends a midi value into [mtof], which converts it to frequency.

[osc~] receives the frequency value and outputs a continuous sine wave.

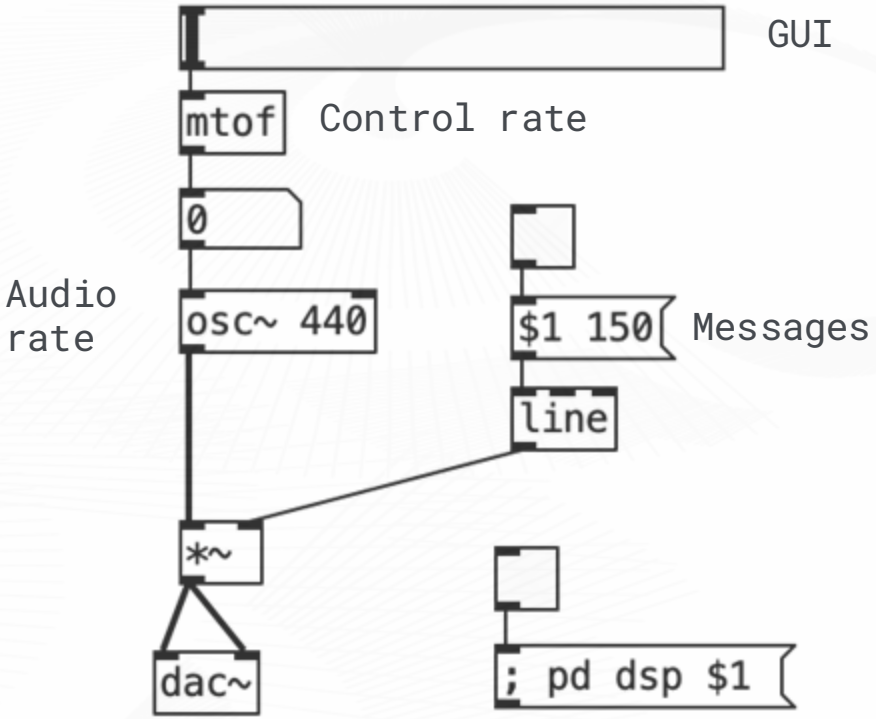
Control the volume of the sine wave (on/off) with the toggle connected to the message [$1 150]

[line] receives the message and creates a smooth amplitude envelope: go to 0 or 1 in 150 milliseconds

[*~] multiplies oscillator output by the envelope generated by [line]

The audio is sent to [dac~] digital to audio converter.

Start and stop Pd with the toggle connected to the message [pd dsp $1] or with the DSP checkbox on the console



Remember: there are two modes: Edit mode and execute mode (you switch between them with Ctrl-E or under Edit -> Edit mode).

Download and install nn~ https://github.com/acids-ircam/nn_tilde/releases/tag/v1.6.0

- Uncompress the .tar.gz file in the Package folder of your Pd installation, i.e. in Documents/Pd/externals/
- Add new path in the Pd/File/Preferences/Path menu pointing to the nn_tilde folder
- MacOS → Terminal cd to nn_tilde folder and run

```
xattr -r -d com.apple.quarantine Documents/Pd/externals/nn_tilde
sudo codesign --deep --force --sign - Documents/Pd/externals/nn_tilde/*.dylib
sudo codesign --deep --force --sign - Documents/Pd/externals/nn_tilde/nn\~.pd_darwin
```

Download course materials

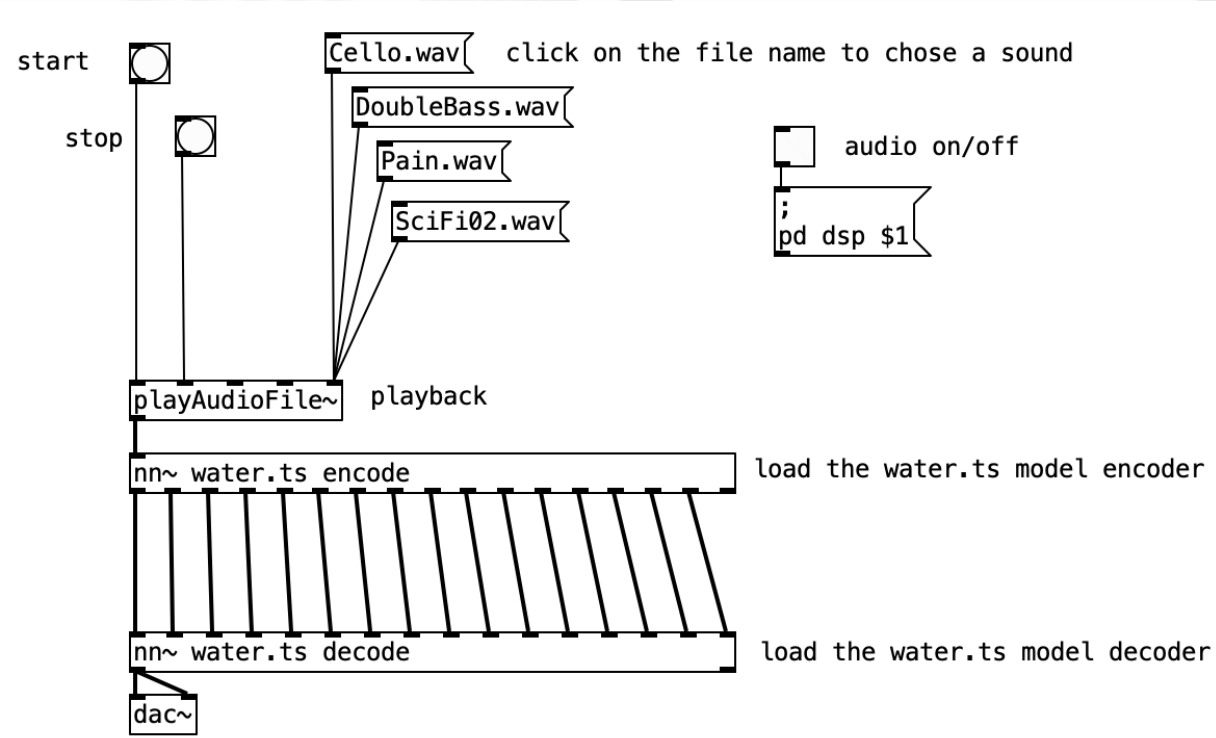
https://drive.google.com/drive/folders/14xMmuIaipBPtCaAKzbML0UnByqoHOGwG

Open MAIN.pd

This patch loads the RAVE model water.ts using the [nn~] object.

- Start Pd by clicking the audio on/off toggle
- Choose a sound and send it into the model
- Click the start button to play the sound

You will hear how RAVE tries to the original sound based on the data used to train the model, in this case audio recordings of water



Some examples of real-life applications:

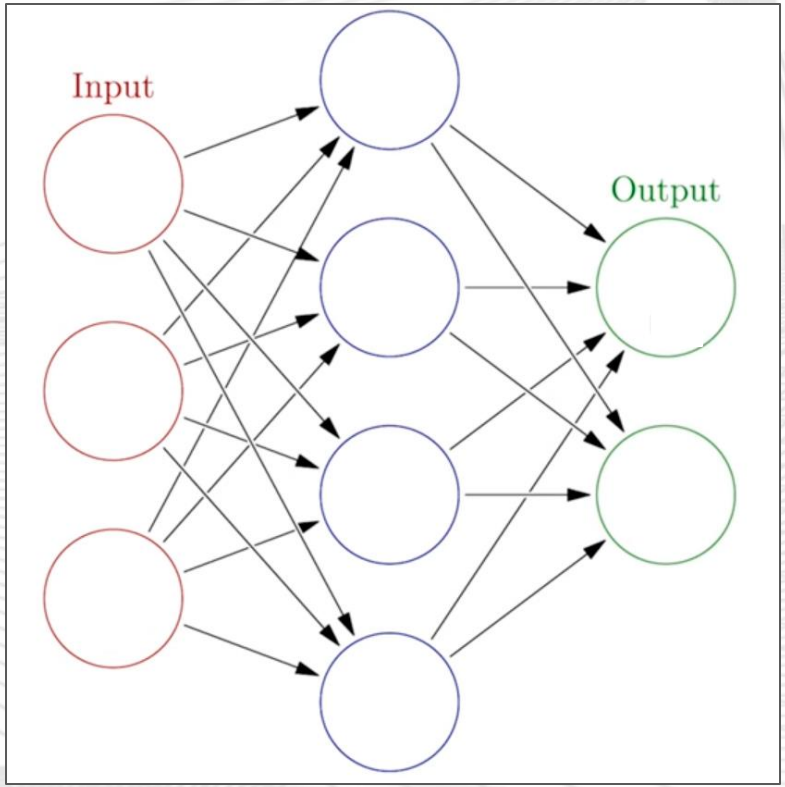
Sound Desing

Music Production

Composition and Sound Art

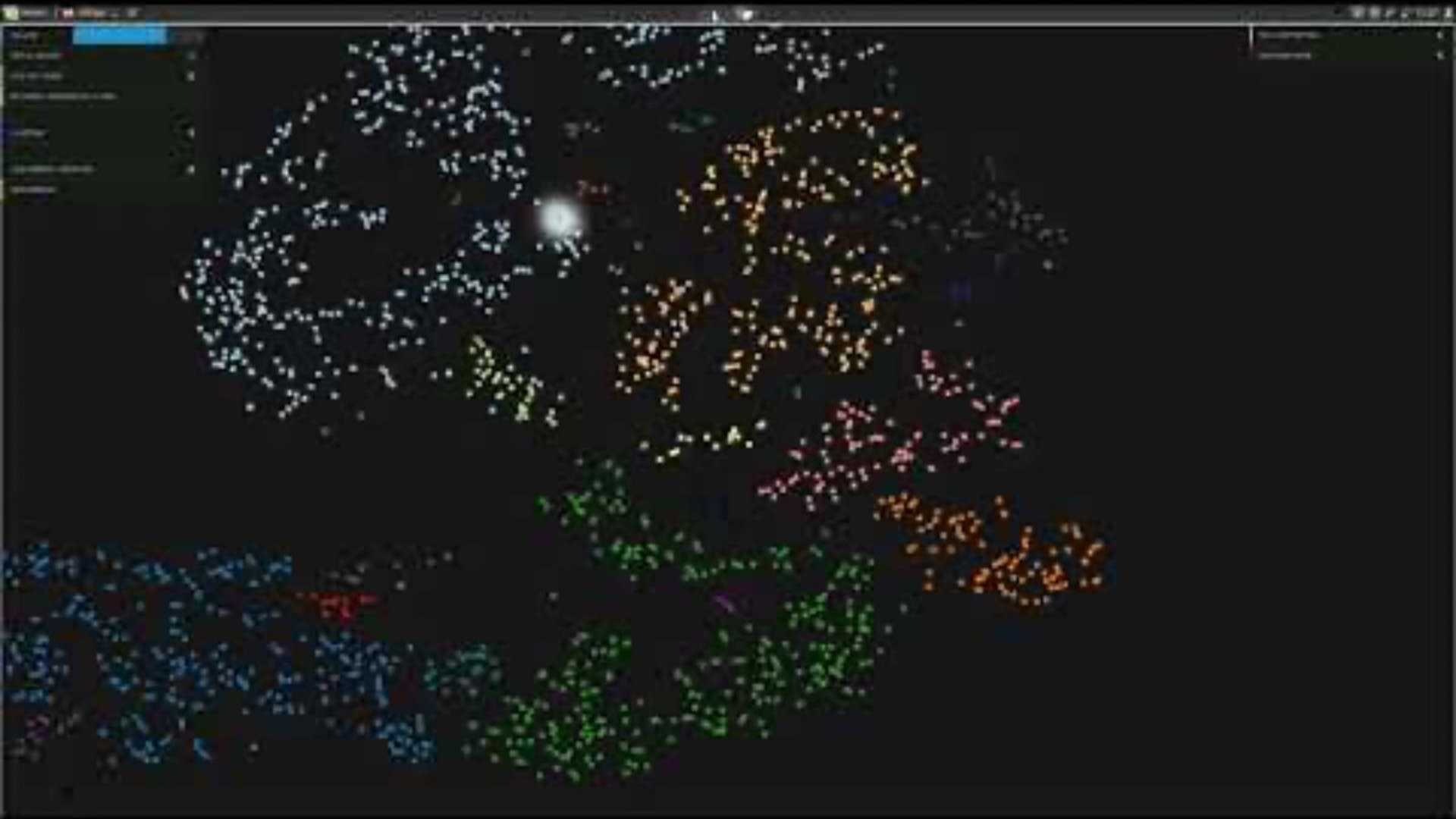
Cracklebox

Sound Classification & Exploration

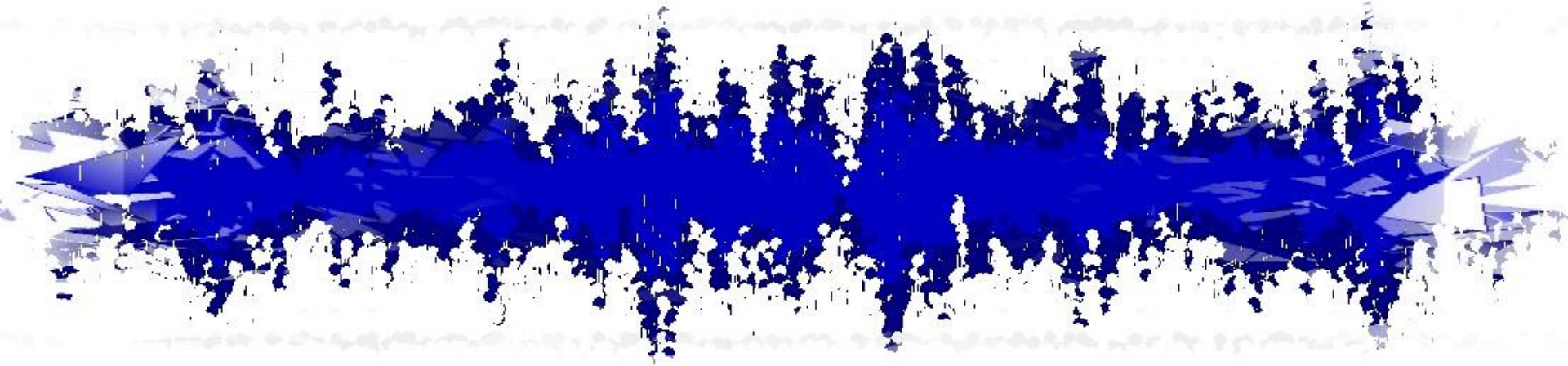


AudioStellar

FluCoMA



Sound Desing




MIX
Refresh
Clear

Music Production

Augmented Instruments Laboratory

iZOTOPE

Composition & Sound Art

Jennifer Walshe

Stephanie Dinkins

DADABOTS

intelligent instruments LAB

Memo Akten

Jonathan Reus

Marco Donnarumma