

Neural Bird Synthesizer

Neural Sound Synthesis Instrument

Xinmeng Li

INTRODUCTION

Neural Bird Synthesizer is a software instrument that explores birdsong through RAVE neural audio synthesis. It began with the birds I heard near my school in the morning and my wish to create bird sounds of my own. Starting from fixed samples, I explored ways to regenerate and combine sounds, developing three modules: Basic, Granular, and Build Your Own Birds. Users can select complete calls, shape granular textures, or blend the latent representations of different calls before decoding them into audio, developing their own combinations of bird sounds.

Motivation and references

During morning walks near my school, I often heard different bird calls and became interested in their sounds. I wanted to create bird sounds of my own, but felt that fixed samples did not offer enough freedom for what I imagined. This led me to look for other approaches to sound synthesis.

I initially imagined making a birdsong piece in Max. As the project developed, I wanted to give users more choice in combining and shaping the sounds themselves. The project shifted from a piece in which I organized the sounds to a software instrument through which other people could explore birdsong.

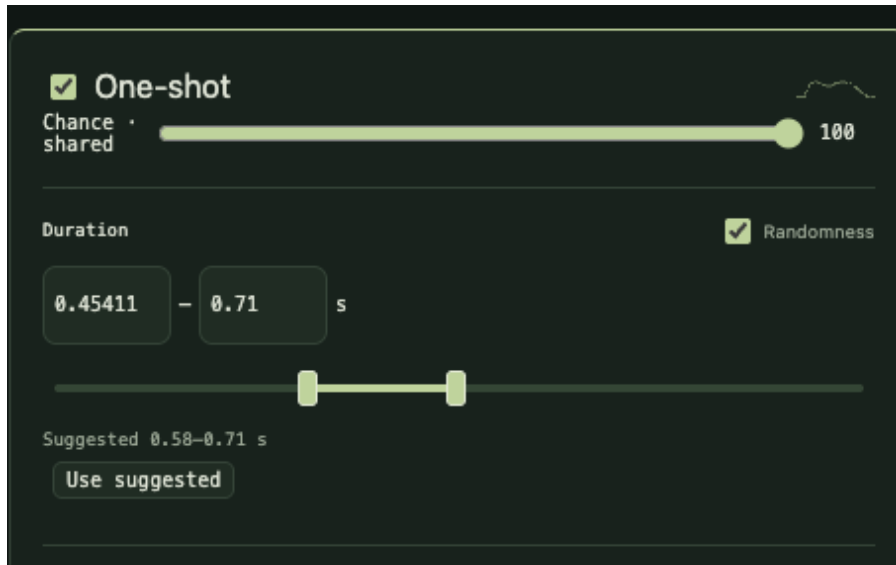
RAVE and sound preparation

I chose RAVE (Realtime Audio Variational autoEncoder), developed by the ACIDS team at IRCAM, as the technical foundation. RAVE is a neural audio synthesis model that learns sound characteristics from training recordings. An encoder converts audio into compact numerical representations, called latent representations, and a decoder reconstructs audible sound from them. Changing or combining these representations offers another way to transform sound. (Caillon and Esling, 2021)

I collected bird-call samples online, reduced noise in the recordings, and trained a RAVE model in Google Colab. For this project, RAVE allowed me to work with birdsong as material that could be regenerated and combined, developing variations through the sound representations learned by the model.

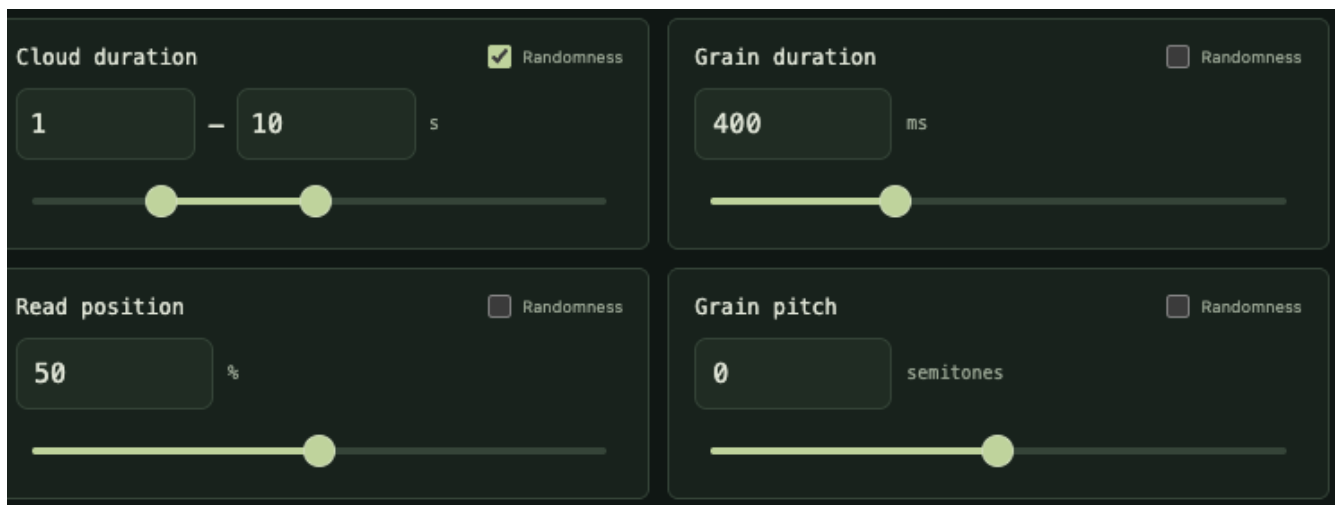
Three modules and their sound workflows

Basic works with complete neurally reconstructed calls. Users select calls and adjust controls such as duration and selection probability, with RAVE decoding producing the sound.



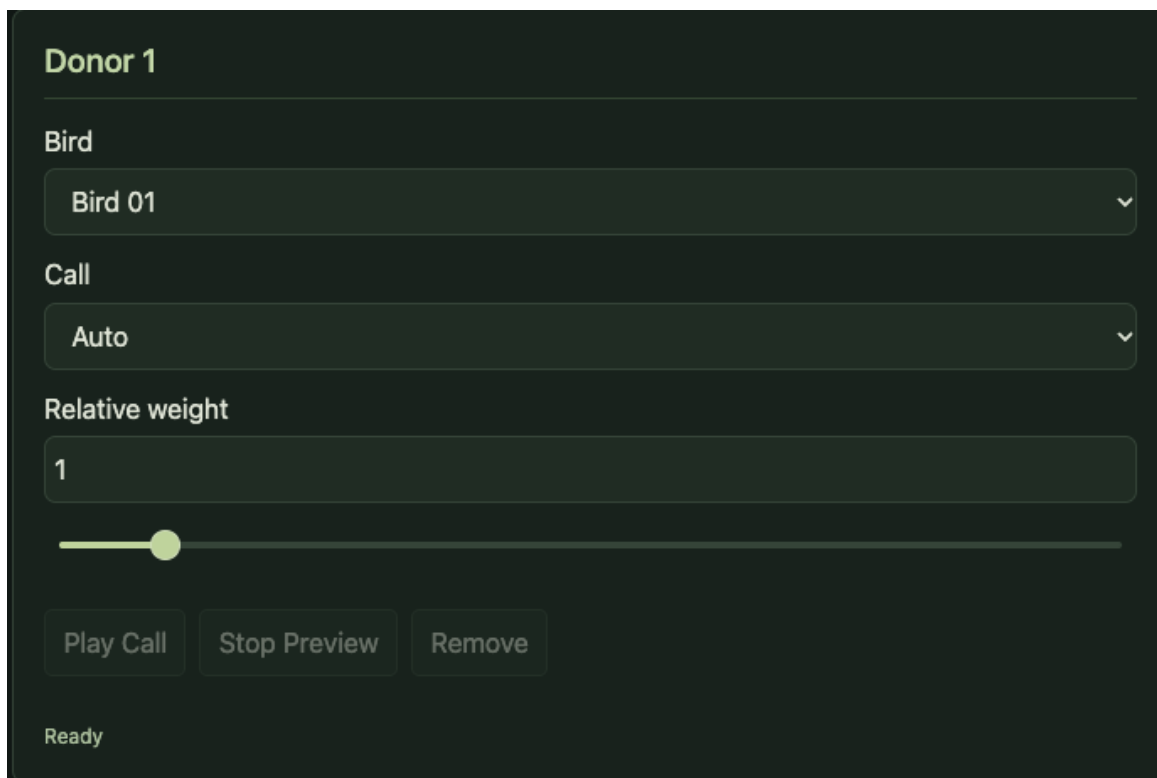
Basic - one-shot selection probability and duration controls (detail).

Granular works with audio that has already been decoded by RAVE. Controls for source regions, grain duration, pitch, and envelopes reorganize that material into fragmented, extended, or overlapping textures. Its grains come from decoded audio; grain playback does not call the neural decoder for each grain.



Granular - cloud and grain duration, read position, and pitch controls (detail).

Build Your Own Birds aligns the latent representations of a main call and donor calls, combines them with adjustable weights, and passes the blend to RAVE for decoding. Users can change how much the materials contribute and compare the result with the main call's reconstruction.



Donor 1

Bird

Bird 01

Call

Auto

Relative weight

1

Play Call Stop Preview Remove

Ready

Build Your Own Birds - donor-call selection and relative weight controls (detail).

A specific design choice: combining calls before decoding

In Build Your Own Birds, I use a main call as the starting point and allow other calls to influence it by adjustable amounts. The materials are mixed at the latent-representation stage before the model generates audio, so the combination takes place before decoding.

I paired this process with sound comparison so users can listen to how the result differs from the main call, then decide how to adjust the contributions of the other materials. This makes the model's internal combination process something users can explore through listening and choice.

Interaction and sonic exploration

I wanted users to be able to retain the choices they made while exploring. Numeric parameter entry, sound comparison, presets, generation history, and WAV export allow different results to be saved and revisited.

The three modules offer different entry points: working with complete calls, reorganizing sound at the grain level, or combining materials through latent representations. Users can choose an approach according to what interests them about the sound.

Process and reflection

Turning sonic ideas into usable software was difficult but compelling. I worked with an LLM coding agent, developing strategies, setting requirements, and deciding the direction of the project while AI assisted with code implementation. I enjoyed asking questions, revising strategies, and gradually turning ideas into functions I could use.

The project changed how I thought about the form of a work. Alongside making music myself, I could design a tool that invites other people to explore sound. My initial interest in morning birdsong developed into a continuing interest in how sounds can be learned, combined, and made available for others to shape.

Contributions and tools

I developed the concept, organized and denoised the bird-call materials, trained the model in Google Colab, and directed the instrument's functional design and development strategy. An LLM coding agent assisted with code implementation. The neural synthesis is based on IRCAM's RAVE.

The main tools and environments include RAVE, Google Colab, Python, PyTorch, FastAPI, and a browser interface.

References

Caillon, Antoine, and Philippe Esling. (2021). RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. arXiv preprint, arXiv:2111.05011.