

# *EVAL* for Commodore 64

Yigit Kolat\*

Seattle, Washington, U.S.A.

yigitkolat@gmail.com

## Abstract

This article introduces *EVAL*, a musical work that explores two contemporary possibilities within a vintage environment — namely the Commodore 64 and its MOS 6581/8580 SID sound chip. One of these possibilities is AI in retrocomputing: a simplified decoder, adapted to the Commodore 64 BASIC environment, generates short melodic lines derived from a pre-trained variational autoencoder. The other involves real-time sound processing and improvisation using a SID emulator. The performance, controlled live via two Nintendo Switch JoyCons, begins as a playful interaction between human and machine and takes an unexpected turn.

## 1 Introduction

### 1.1 Motivation

The project explores the Commodore 64 (C64) and its MOS 6581/8580 SID (Sound Interface Device) sound chip within a contemporary computational, conceptual, and artistic framework. It follows several objectives: integrating generative AI in audio/music into a vintage computing environment, using the SID chip in the context of real-time sound processing, and bringing these two aspects together within the framework of a musical composition/performance.

While various tools exist for musicians to experiment with the SID’s unique soundscape, a programmatic, real-time, and modular approach has been lacking. This gap is addressed with *SID0sc*, a custom SuperCollider Unit Generator (UGen). The generative AI component of the project, as discussed in the following sections, emerges from both technical and artistic limitations. Echoing the ethos of the C64 demoscene, the project explores how the functionality of a variational autoencoder can be approximated within the constraints of the Commodore 64.

## 2 Background

### 2.1 Retrocomputing, SID, and AI

Retrocomputing is a thriving subculture that attracts programmers interested in the challenge of creative programming within limited resources. Introduced in 1982 and discontinued in 1994, the Commodore 64 has provided a versatile environment for the programmers. The system’s MOS 6581/8580 SID sound chip offers a characteristic soundscape with its four waveforms, three tone generators, and three envelope generators, among other components.

The appearance of cross platform emulators such as VICE enabled musicians and programmers to explore the C64 without original hardware. VICE notably features *reSID*, a robust SID emulator developed by Dag Lem and available as open-source software (Lem, 2023).

There is an existing body of literature on artificial intelligence within the context of 1980s personal computing, with a particular focus on the C64. Examples include rule-based systems (Brain and Brain, 1984) as well as experiments with artificial neural networks (Walker, 1987). Recently,

---

\*Independent Artist/Researcher

artificial intelligence paradigms have re-emerged in retrocomputing, typically involving computations performed externally due to hardware constraints (Bild, 2024).

## 2.2 Related works

Nick Bild's Commodore 64 AI Image Generator significantly influenced this project (Bild, 2024). Bild uses a probabilistic principal component analysis algorithm to obtain a set of weights, mean values, and a covariance value to scale the positions of sprites on the screen. The PPCA is trained with 94 hand-made sprites.

Alexander Sashnov's AY-8912 emulation library and its AY UGen for SuperCollider provided additional inspiration. The library emulates the General Instrument AY-3-8910 soundchip used in vintage systems like ZX-Spectrum 128 and Atari ST.

## 3 About the work

### 3.1 System description

#### 3.1.1 Usage of AI

The work features "machine" and "human" interactions with the SID chip. The machine interaction utilizes a variational autoencoder (VAE) trained on 94 arrays of 64 binary values, each representing a monophonic melody formatted as an 8x8 grid, where the x-axis corresponds to time and the y-axis to pitch. After training the VAE, decoder weights consisting of an 8x64 matrix and a 64-element bias are stored. All values are rounded to the nearest tenth.

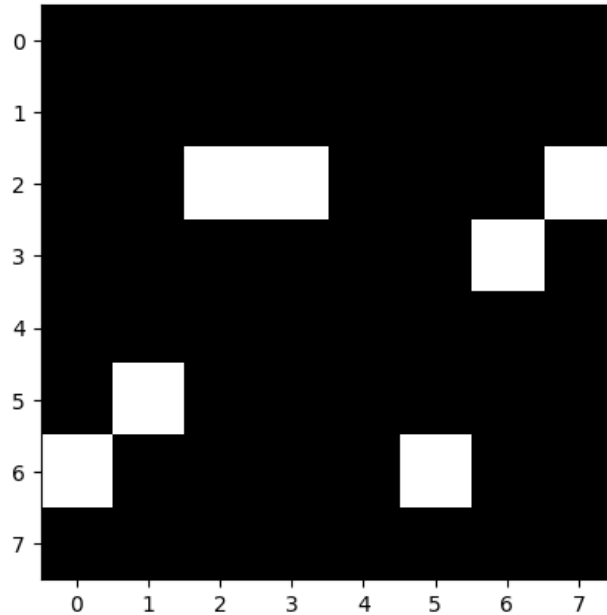


Figure 1: A 8x8 representation of a melody taken from the training set.

Post-training, a simplified decoder implemented in C64 BASIC generates melodic "prototypes" from stored weights and biases. We find this method to be an effective compromise, situated between the ambitious task of training a variational autoencoder directly within a BASIC environment and the simpler alternative of embedding a static, pre-computed prototype as a BASIC array.

The decoding utilizes a fixed latent vector  $z$  which contains 8 values. The prototype is calculated via  $x = \sigma(W \cdot z + b)$ , where  $W$  and  $b$  are the weight matrix and bias array, and  $\sigma$  is the sigmoid activation function which maps each of the 64 elements into  $[0, 1]$ . For C64 BASIC, we use an accommodating piecewise-linear approximation for the sigmoid function:

$$s(x) = \begin{cases} 0, & \text{if } x \leq -2 \\ \frac{x+2}{4}, & \text{if } -2 < x < 2 \\ 1, & \text{if } x \geq 2 \end{cases}$$

```

640 SUM = SUM + Z(J) * WDEC(IDX)
650 NEXT J
660 IF SUM <= -2 THEN S = 0
670 IF SUM >= 2 THEN S = 1
680 S = (SUM + 2) / 4
690 IF S > 0.5 THEN PAT(I) = 1
700 IF S <= 0.5 THEN PAT(I) = 0
710 NEXT I

```

Figure 2: Piecewise-linear approximation for the sigmoid function and its implementation in C64 BASIC

The prototype is then linearly mapped into an arbitrary frequency range by applying a piecewise transformation: for each of the 8 columns we extract the row index  $n_t \in \{0, \dots, 7\}$  of the "1 bit", normalize it to  $\frac{n_t}{7}$ , scale and offset to the intended SID register span via

$$F = \left\lfloor \frac{n_t}{7} \times 80 + 160 \right\rfloor,$$

and finally write (or POKE) the 16-value  $F$  into SID's frequency registers.

### 3.1.2 Real-time usage of SID

The "human interaction" aspect employs SID0sc, a SuperCollider UGen developed by the author. Essentially a wrapper around Dag Lem's reSID emulator, the UGen is capable of providing real-time oscillator behavior via the emulation of SID's four oscillators –triangle wave, sawtooth wave, pulse, and noise. For the live processing, the UGen is enriched through nonlinear sample feedback using Fb1, a UGen dedicated to single sample feedback and feedforward operations. This environment provides a rich timbral soundscape, which is then explored by a number of control parameters linked to two Nintendo Switch JoyCon controllers via OSC messaging over Bluetooth transmission via joycon-osc library.

Two JoyCons control two independent streams of events. The left JoyCon adjusts feedback via stick movements, where vertical and horizontal motions of the stick are mapped to the pitch and amplitude values. Similar to the conditions of a no-input mixer feedback, the amplitude argument does not affect the overall amplitude, but drastically changes the timbral profile of the sound. The overall amplitude this particular stream is manipulated via the accelerometers of the left controller. The right JoyCon manages additive synthesis across four SID oscillators, modulating their independent amplitudes through accelerometric data.

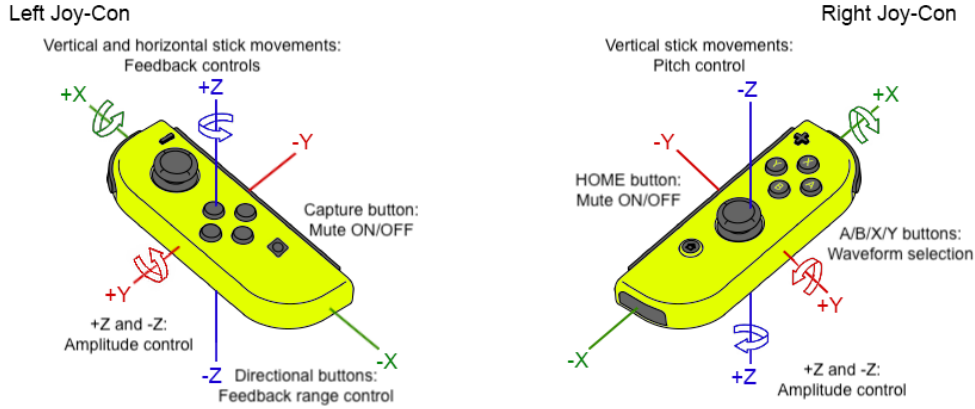


Figure 3: Two JoyCons and their functions.

## 3.2 Narrative

*EVAL* begins as a playful interaction: the AI presents a monophonic melody, urging the performer's improvisational response. The performer has complete freedom in their decisions. However, each

attempt is met with failure. Initially, this is communicated in a supportive tone, but the prompts gradually escalate in severity. Ultimately, the scenario is revealed to be a rigged artistic evaluation within a fictional authoritarian government, orchestrated by a committee overseeing cultural affairs.

The visual component of the work, written in 6502 assembly using KickAssembler utilizes effects such as raster bars that are characteristic to the Commodore 64 aesthetic (see Figure 4).



Figure 4: A screenshot from the visual component of the work.

## 4 Future work

Currently, no real-time evaluative feedback loop exists between the AI and performer, which is consistent with the piece's narrative where an evaluation with a predetermined conclusion is being conducted. Future iterations aim to incorporate genuine interaction and evaluative processes, exploring deeper human-machine dynamics. Additionally, implementing the fixed-media aspects on actual Commodore 64 hardware remains a compelling direction.

## References

- Bild, N. (2024). Commodore 64 AI image generator. Available at [https://github.com/nickbild/c64\\_gen\\_ai](https://github.com/nickbild/c64_gen_ai). Last accessed 13 April 2025.
- Brain, K. and Brain, S. (1984). *Artificial Intelligence on the Commodore 64*. Sunshine Books.
- Lem, D. (2023). reSID: MOS 6581 / 8580 SID software emulation. Available at <https://github.com/daglem/reSID>. Last accessed 13 April 2025.
- Walker, J. (1987). Neuron network associative memory program. Available at <https://www.fourmilab.ch/documents/commodore/BrainSim/>. Last accessed 13 April 2025.