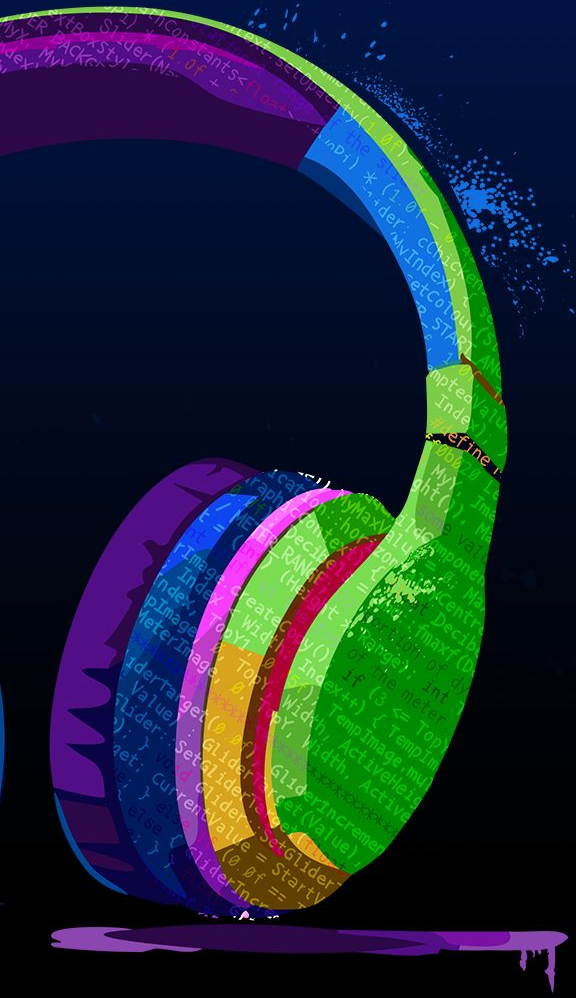


ADC24 Bristol

PRACTICAL LESSONS LEARNED FROM BUILDING AUDIO APPLICATIONS USING AI MODELS

CHANG HUN SUNG

About Supertone

SUPERTONE

MARG
MUSIC & AUDIO RESEARCH GROUP

NANSY

Neural analysis & synthesis
Unified voice synthesis framework

About Speaker

- Previously Game Engine Programmer. Used to develop GPU Graphics pipeline or Physics Engine with C++.
- Joined Supertone at 2022. Developer of User-centric products.
- Currently the Leader of the **'AudioDev team**'

SUPERTONE™

The Voice Intelligence Platform

Application
Web / Local

CLEAR
VOICE SEPARATOR
Real-Time Voice Separator for Voice Separation

SHIFT
REAL-TIME VOICE CHANGER
Real-Time Voice Changer for Interactive Content Creation

Project
SCREENPLAY
Text to Speech

Project
VOICE MARKET PLACE

VOICE LIBRARY

Solution

API
TTS
CVC
SVS

SDK
RTVC
RTSE

Framework

NANSY
Neural analysis & synthesis
Unified voice synthesis framework

NAUD
Efficient development of voice AI-based applications

Introduction

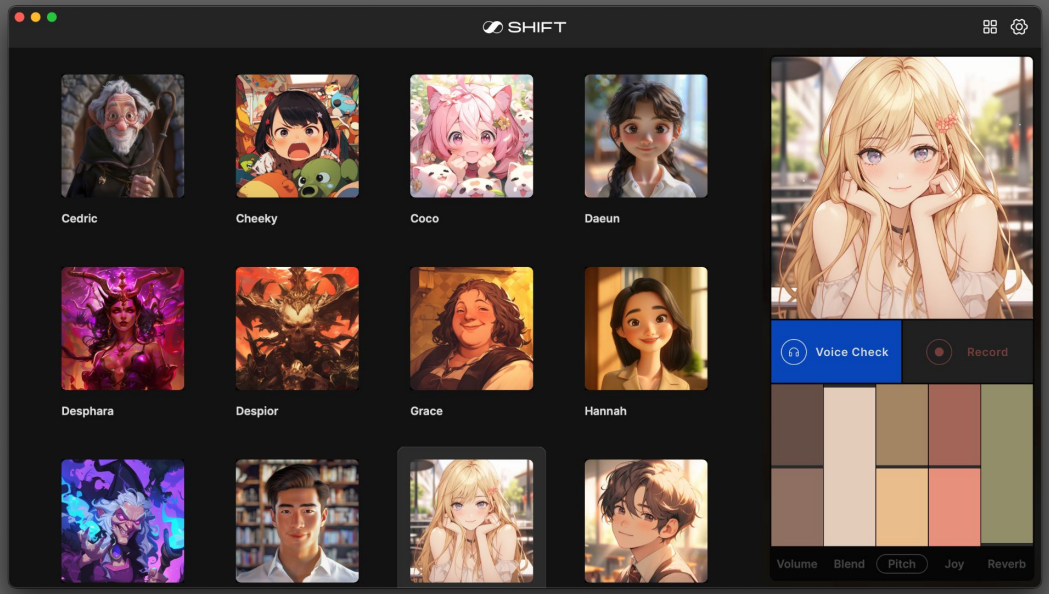
Quick Demo

Supertone Clear

Real-time voice separator Plug-in

Supertone Shift

Real-time voice conversion Software



SUPERTONE™

Supertone Air

Reverb & EQ Dialogue Match


SUPERTONE
AIR
REVERB & EQ
DIALOGUE
MATCH
R


SUPERTONE
AIR
REVERB & EQ
DIALOGUE
MATCH
Capture the Air in Dialogue

Deep Dive

Supertone Clear

Real-time Speech Enhancement

Data preprocessing to obtain clean voice

Recorded Voice with Noise → RTSE →

NANSY

Neural analysis & synthesis
Unified voice synthesis framework

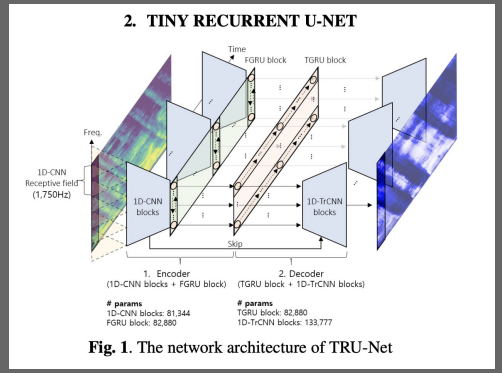


Fig. 1. The network architecture of TRU-Net

REAL-TIME DENOISING AND DEREVERBERATION WITH TINY RECURRENT U-NET *H, Choi et al*

- Lightweight enough to run in real-time
- Effectively separates non-voice and wet reverberant voice.
- Great starter to experiencing a Product development.

Developing a Plug-in

Definitely Fun

Drawing a prototype

Experimental Features

Pricing a plug-in

Mostly Fun

Writing a code

Writing a tests

Automating build pipeline

Supports variety of platforms

Not Fun at all

Worrying about crack

Worrying about crash

Worrying about...

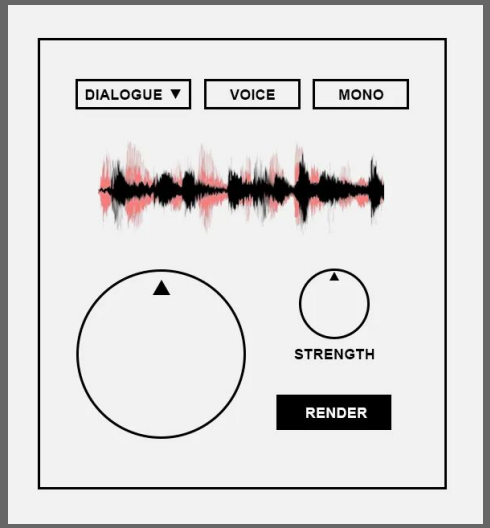
.

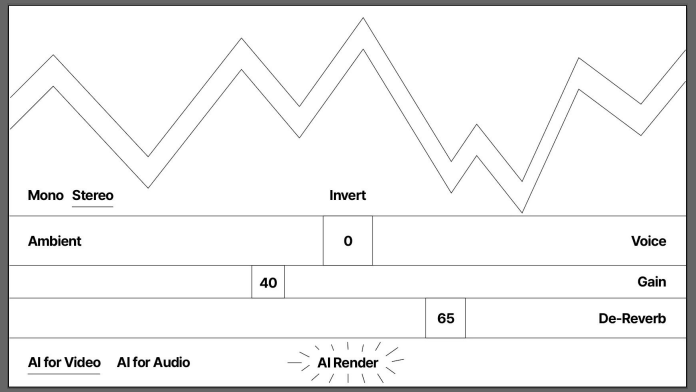
.

.

Developing a Plug-in

Definitely fun part





Developing a Plug-in

Definitely fun part

GOYO(beta) / Sep 2022

- Apply fully Physically Based 3D Rendering with OpenGL
- Implements dynamic deferred lighting.
- Large memory, GPU consumption

Supertone Clear / Sep 2023

- Compact and restrained graphic.
- Focused on User Experience .
- More informative

Developing a Plug-in

Not Fun at all

Waiting for bottleneck…

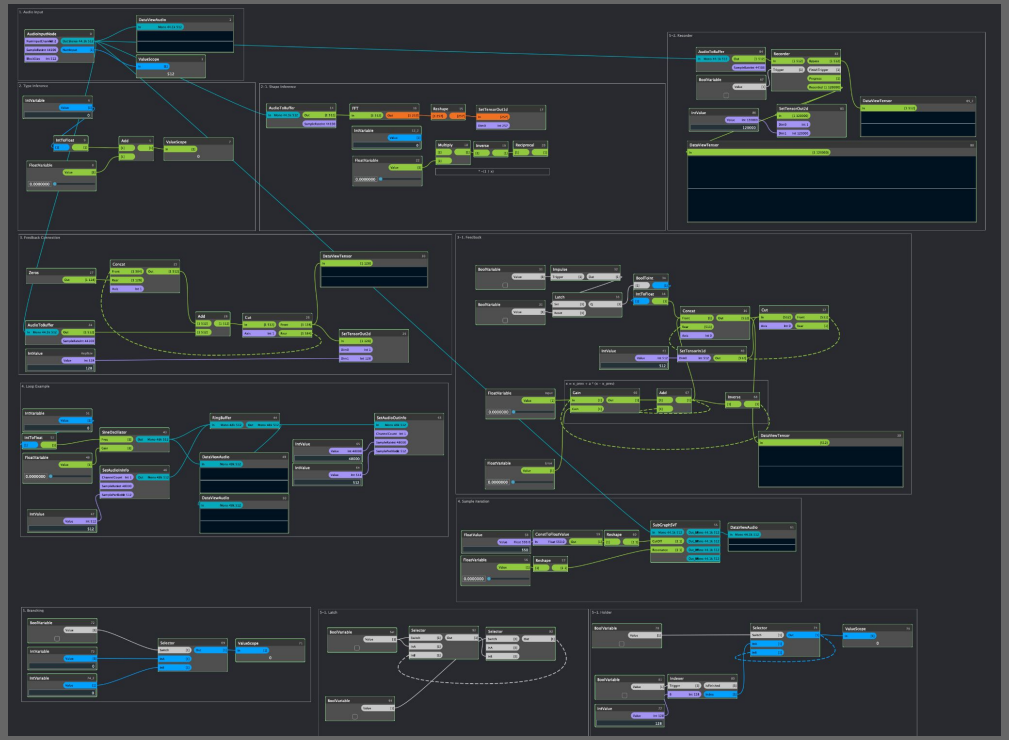
Research → MLE → SDK →

Bottleneck Bottleneck Bottleneck

Deep Dive

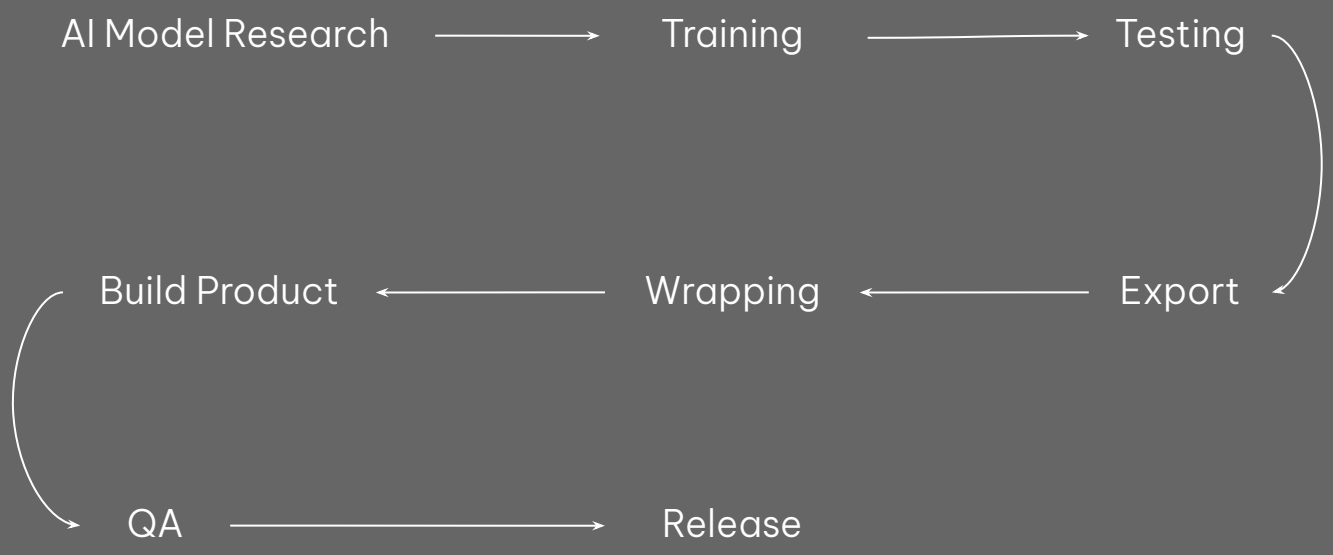
Naud Framework

SUPERTONE™



- Visual Programming Tool
- Inference Engine Integration
- Support Audio / MIDI IO
- Performance-Oriented
- Modular design

Launching an AI-powered audio software



How does it change the pipeline?

AI Model Research → Training

Profiling

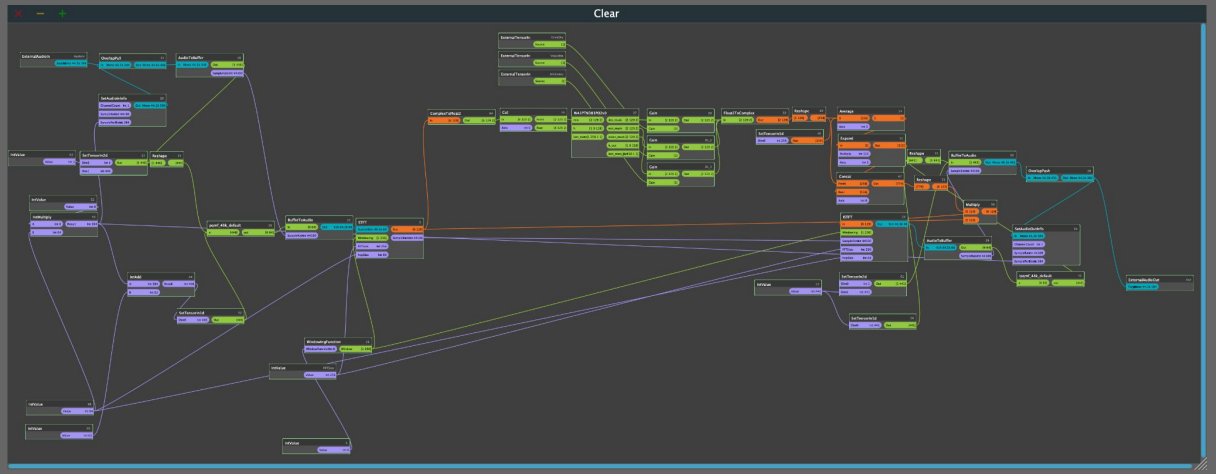
Experiment

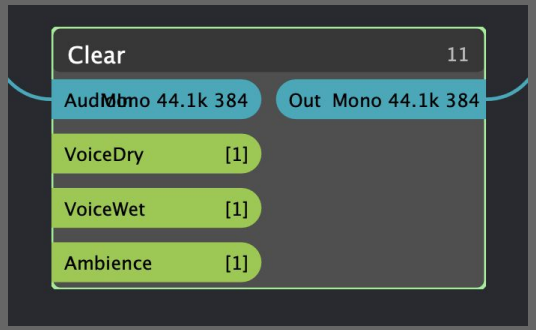
Parameter Design

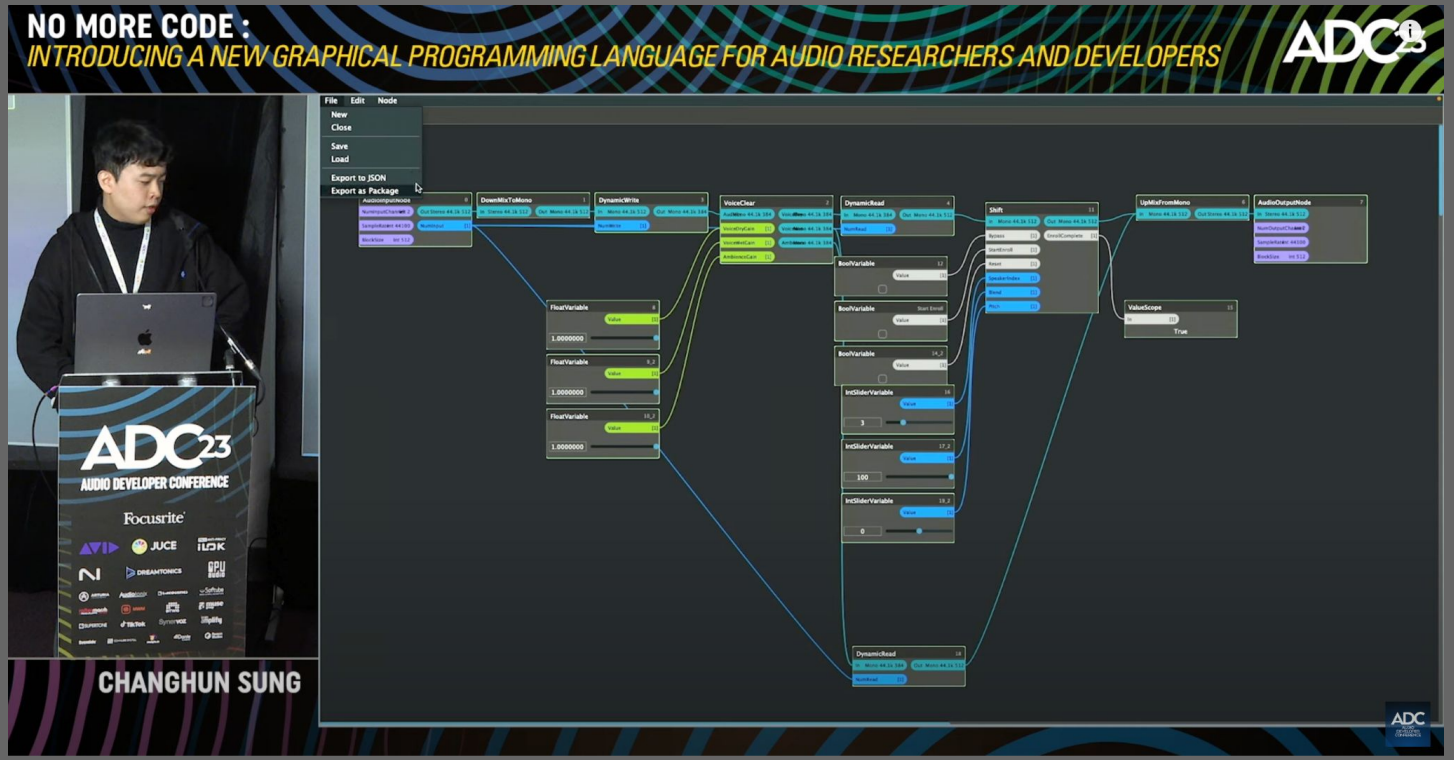
QA

Attach UI → Release

How Clear was made







ADC23 *NO MORE CODE: New Graphical Programming Language for Audio Research and Developers*

Community Beta Release

Mar. 2024

Stay tuned.

Community Beta Release

Mar. 2024

Failed!

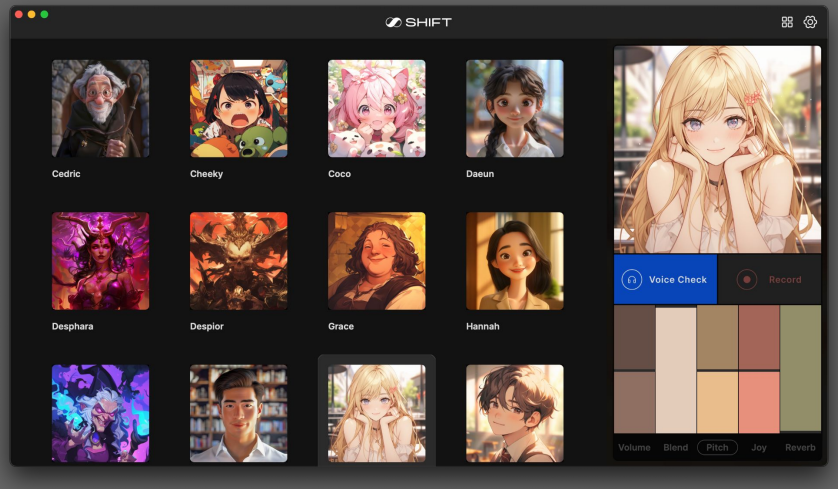
Stay tuned.

Deep Dive

Supertone Shift

Supertone Shift

Real-time voice conversion Software



- Desktop Application runs on Windows and macOS
- Can connect with Google Meet, Discord, Zoom etc.
- Provides variety of persona.

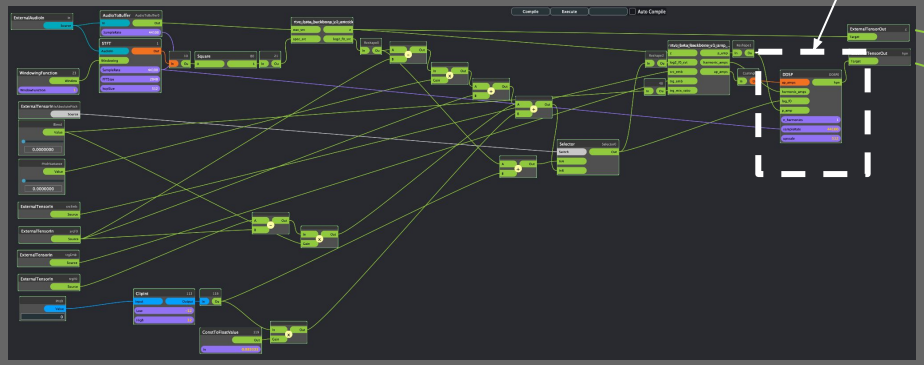
- Utilize DDSP to process real-time audio.
- Provides Pitch, Pitch variance and Blend.
- Runs on CPU in with 35ms latency.

Developing Supertone Shift

With using Naud framework

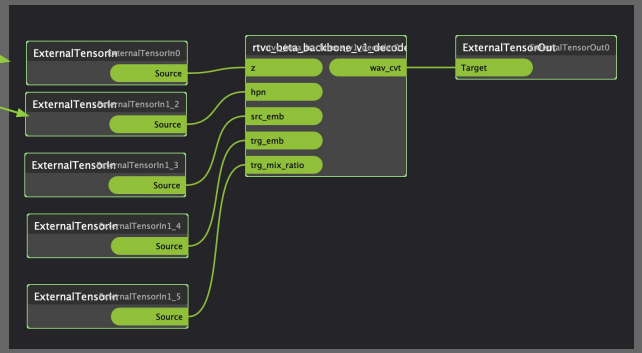
DDSP node

Encoder



Input voice analysis

Decoder



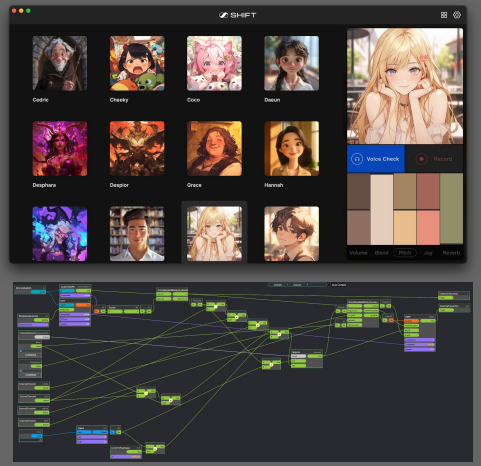
Output voice synthesis

SUPERTONE™

Developing Supertone Shift

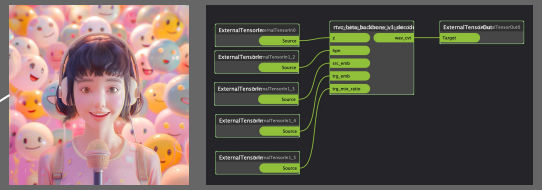
With using Naud framework

Encoder

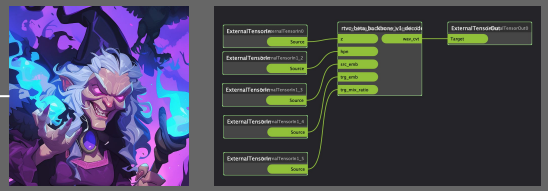


Input voice analysis

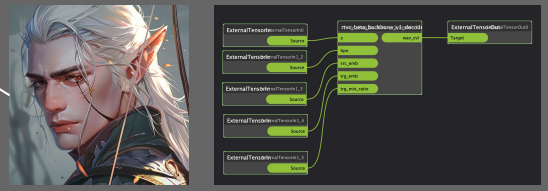
Decoder



Decoder



Decoder

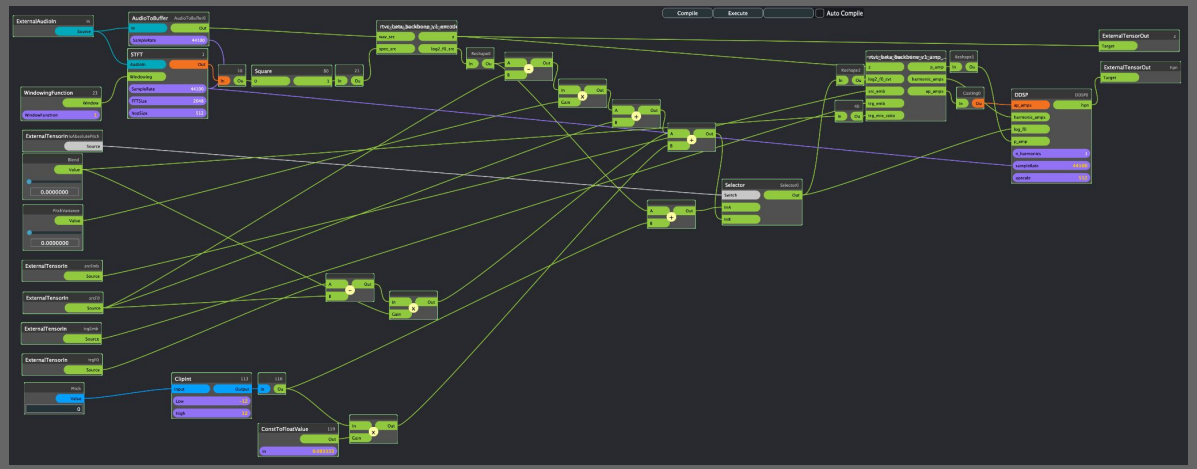


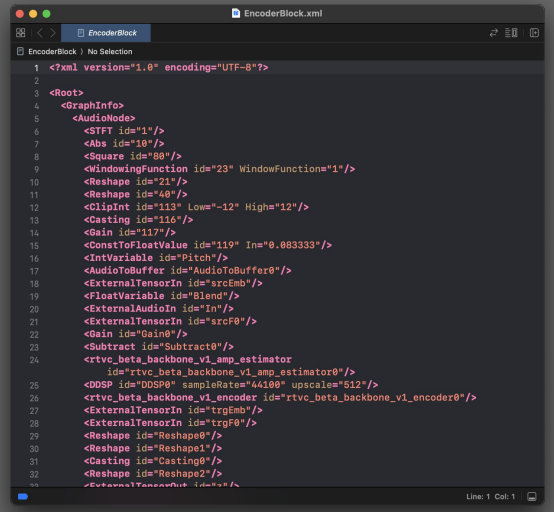
Output voice synthesis

SUPERTONE™

Developing Supertone Shift

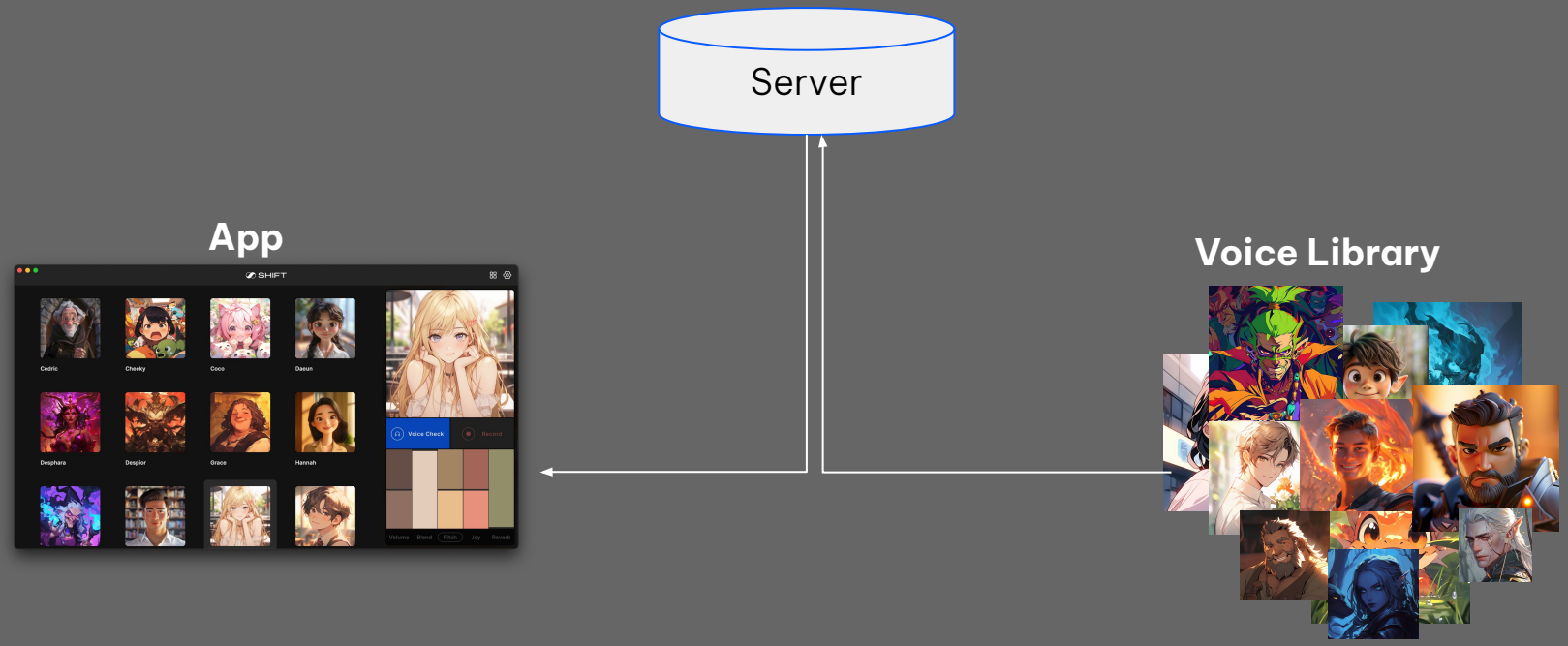
With using Naud framework





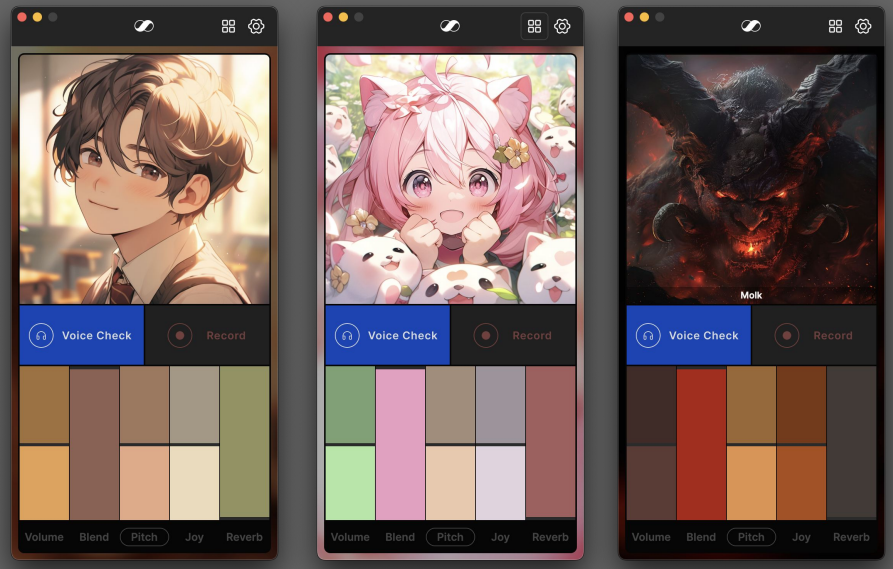
Developing Supertone Shift

With using Naud framework



Developing Supertone Shift

Who is user?



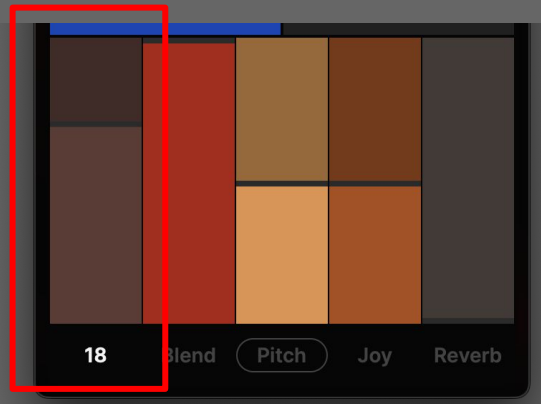
- Mostly non professional for audio.
  - VTuber
  - Podcaster
  - Voice contents creator
- Not familiar with traditional audio software.

Developing Supertone Shift

Volume slider problem

-24dB ~ +24dB vs **1 ~ 25**

- It is standard representation of volume.
- It is still intuitive representation.
- It is not that important thing, why should we argue with this?



- Why should our users have to know about standard?
- There is more intuitive representation.
- Because it matters whole experiences!

Deep Dive

Supertone Air

SUPERTONE™

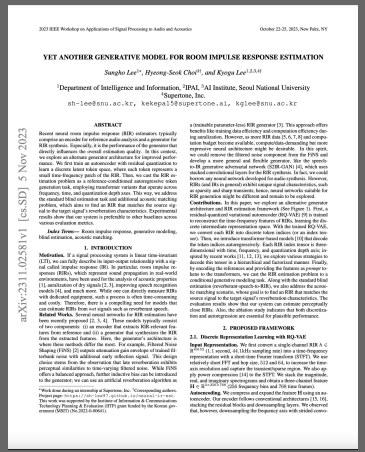
Supertone Air

Reverb & EQ Dialogue Match

Developing Supertone Air

Implementing Transformer on Naud Framework

Yet Another Generative Model For Room Impulse Response Estimation *S, Lee et al*



2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics — October 22-25, 2023, New Paltz, NY

arXiv:2311.02581v1 [eess.SP] 5 Nov 2023

YET ANOTHER GENERATIVE MODEL FOR ROOM IMPULSE RESPONSE ESTIMATION

Sungho Lee[1*], *Hyeong-Seok Choi*[1†], *and Kyogu Lee*[1,2,3,4†]

[1]Department of Intelligence and Information, [2]IPAI, [3]AI Institute, Seoul National University
[4]Supertone, Inc.
sh-lee@snu.ac.kr, kekepa15@supertone.ai, kglee@snu.ac.kr

ABSTRACT

Recent neural room impulse response (RIR) estimators typically employ a single-stage encoder-decoder architecture to directly map reverberant RIR synthesis. Especially, it is the performance of the generator that directly influences the overall estimation quality. In this context, we explore an alternate generator architecture for improved performance. We first train an autoencoder with residual quantization to learn a discrete latent token space, where each token represents a small time-frequency patch of the RIR. Then, we can the RIR estimation task as a conditional token generation task, employing transformer variants that operate across frequency, time, and quantization-depth axes. This way, we address the standard blind estimation task and additional acoustic matching problem, which aims to find an RIR that matches the source signal to the target signal's reverberation characteristics. Experimental results show that our system is performite to other baselines across various evaluation metrics.

Index Terms— Room impulse response, generative modeling, blind estimation, acoustic matching.

1. INTRODUCTION

Motivation. If a signal processing system is linear time-invariant (LTI), we can fully describe its input-output relationship with a signal called impulse response (IR). In particular, room impulse responses (RIRs), which represent sound propagation in real-world environments, are crucial for applications such as speech enhancement [1], sonification of dry signals [2, 3], improving speech recognition models [4], and much more. While one can directly measure RIRs with dedicated equipment, such a process is often time-consuming and costly. Therefore, there is a compelling need for models that can estimate RIRs from sot signals such as reverberant speech.

Related Works. Several neural networks for RIR estimation have been proposed [5, 6, 7, 8, 9, 10, 11, 12]. Typically, they consist of two components: (i) an encoder that extracts RIR-relevant features from reference and (ii) a generator that synthesizes the RIR from the extracted features. Here, the generator's architecture is where these methods differ the most. For example, Filtered Noise Shaping (FNS) [2] outputs attenuation gain envelope of trained filtrerbank noise with additional early reflections signal. This design choice allows the generator to produce RIRs that exhibit similar perceptual similarities to time-varying filtered noise. While FNS offers a balanced approach, further inductive bias can be introduced to the generator; we can use an artificial reverberation algorithm as a (trainable parameter-less) RIR generator [3]. This approach offers benefits like training data efficiency and computation efficiency during serialization. However, as more RIR data [5, 6, 7, 8] and computation budget become available, compute/data-demanding but more expressive neural architecture might be desirable. In this spirit, we could remove the filtered noise component from the FNS and develop a more general and flexible generator, like the speech-to-RIR (S2R) generator proposed in [9]. The S2R U-Net [9] has stacked convolutional layers for the RIR synthesis. In fact, we could borrow any neural network developed for audio synthesis. However, RIRs (and IRs in general) exhibit unique signal characteristics, such as sparsity and sharp transients; hence, neural networks suitable for RIR generation might be different and remain to be explored.

Contributions. In this paper, we explore an alternative generator architecture for improved performance. Specifically, we train a residual quantized variational autoencoder (RQ-VAE) [13] is trained to reconstruct the time-frequency features of RIRs, learning the discrete intermediate representation space. With the trained RQ-VAE, we convert each RIR into discrete token indices (or an index tensor). Then, we introduce transformer based models [14] that decode the token indices autoregressively. Each RIR index tensor is three-dimensional, and we have three axes to consider: time, frequency, and quantization depth. Along with the standard blind estimation (reverberant-speech-to-RIR), we also address the acoustic matching task, which aims to find an RIR that matches the source signal to the target signal's reverberation characteristics. The evaluation results show that our system can estimate perceptually close RIRs. Also, the ablation study indicates that both discretization and autoregression are essential for plausible performance.

2. PROPOSED FRAMEWORK

2.1. Discrete Representation Learning with RQ-VAE

Input Representation. We first convert a single-channel RIR $h \in \mathbb{R}^{1\times N}$ (1 second, 44.1kHz sampling rate) into a time-frequency representation with a short-time Fourier transform (STFT). We use a Hann window of 1024 samples, 256 hop length, and 1024-point fast-axis resolution and capture the transient/sparse region. We also apply power compression [14] to the STFT. We stack the magnitude, real, and imaginary spectrograms and obtain a three-channel feature $\mathbf{H} \in \mathbb{R}^{3\times 513\times T_0}$ (513 frequency bins and 700 time frames).

Autoencoding. We compress and expand the feature $\mathbf{H}$ using an autoencoder. Our encoder follows conventional architectures [15, 16], stacking the residual blocks and downsampling layers. We observed that, however, downsampling the frequency axis with strided convo-

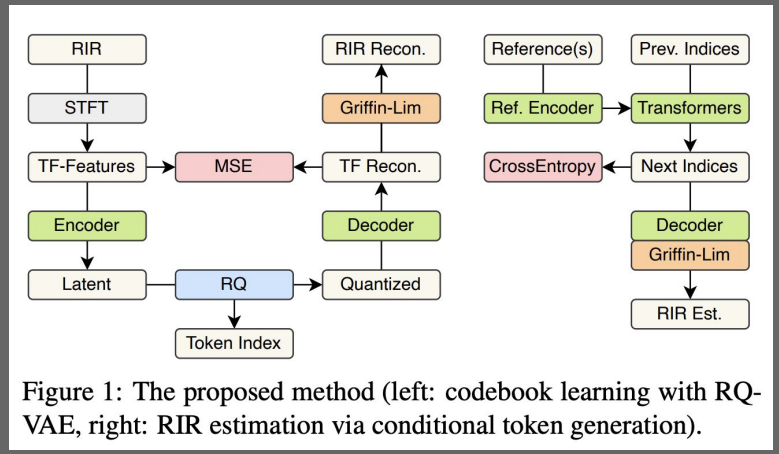
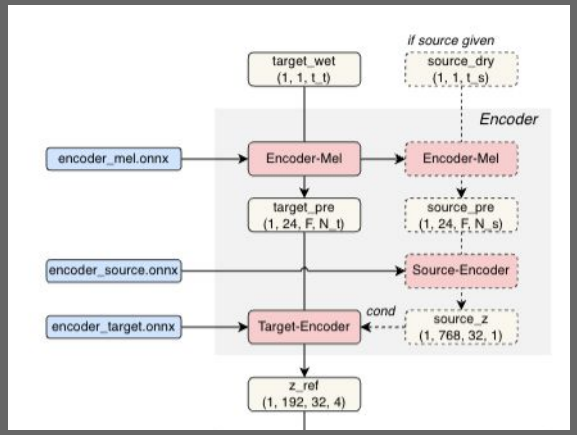


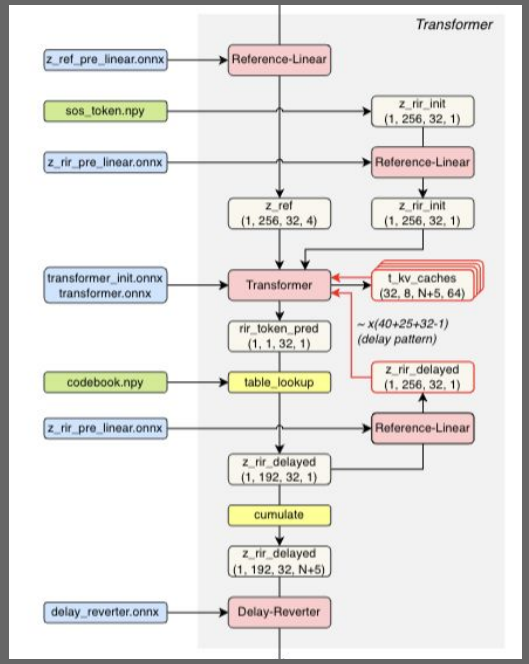
Figure 1: The proposed method (left: codebook learning with RQ-VAE, right: RIR estimation via conditional token generation).

Developing Supertone Air

Inference pipeline of AST model



Encoder



Transformer

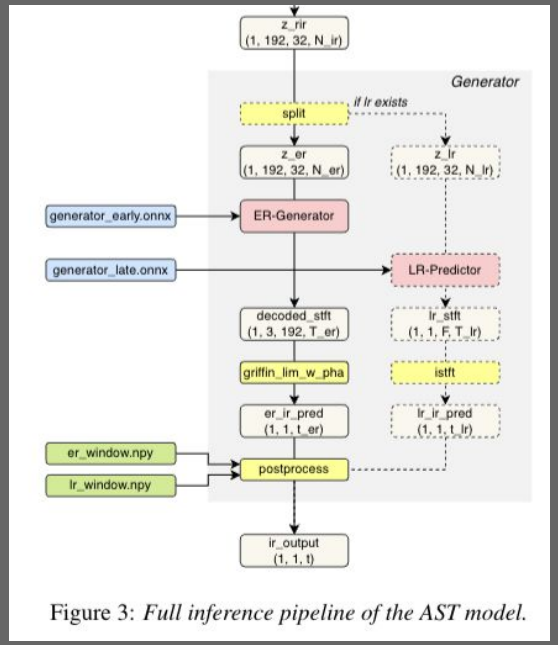


Figure 3: *Full inference pipeline of the AST model.*

Post-processor

Developing Supertone Air

Implementing Transformer with Naud Framework

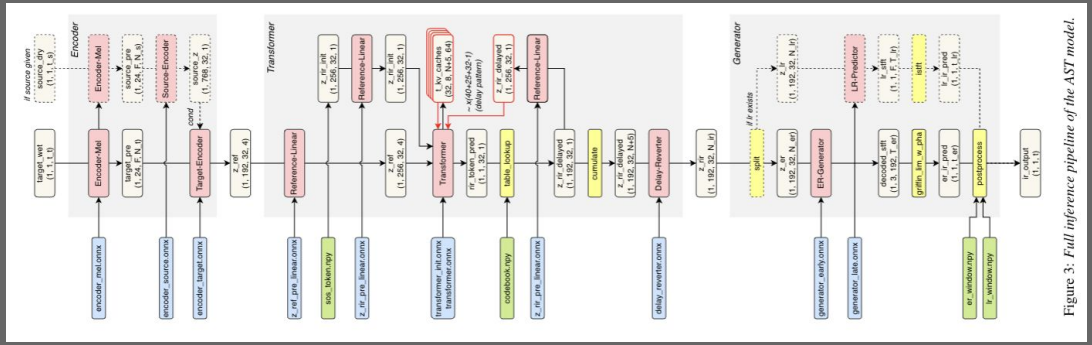
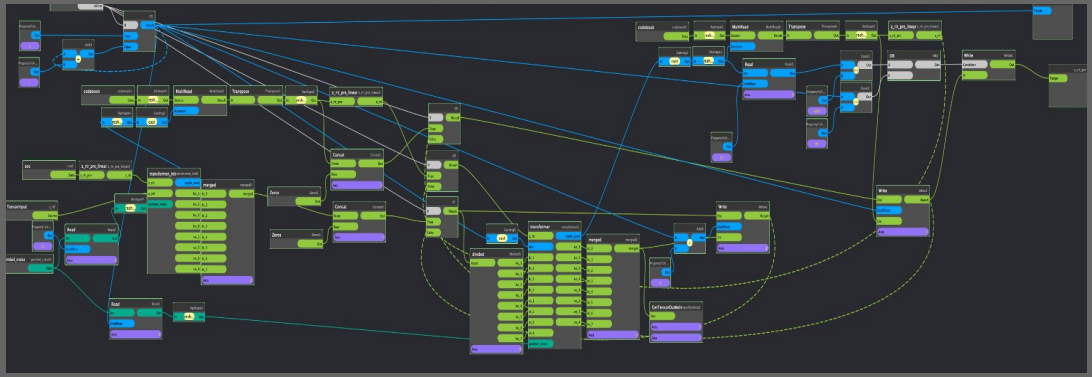


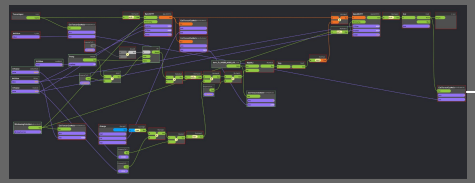
Figure 3: *Full inference pipeline of the AST model.*

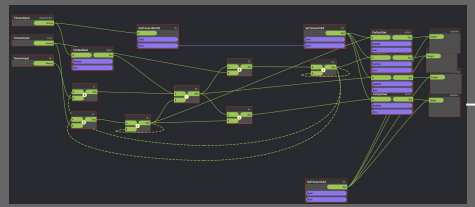


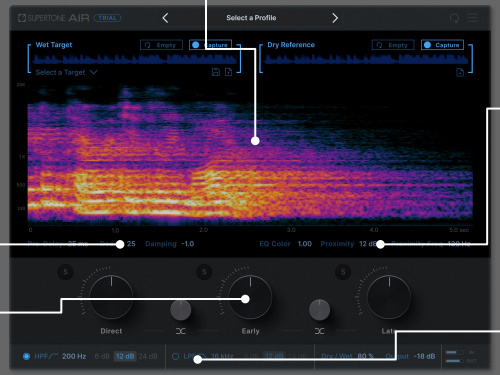
SUPERTONE™

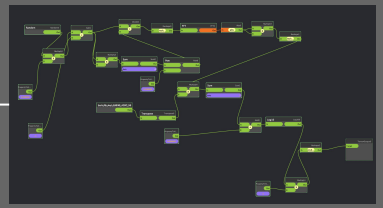
Developing Supertone Air

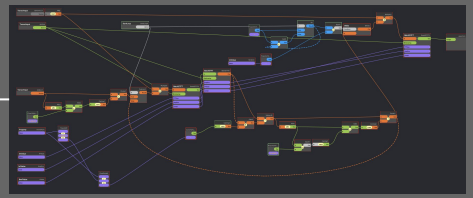
Implementing DSP algorithms with Naud Framework





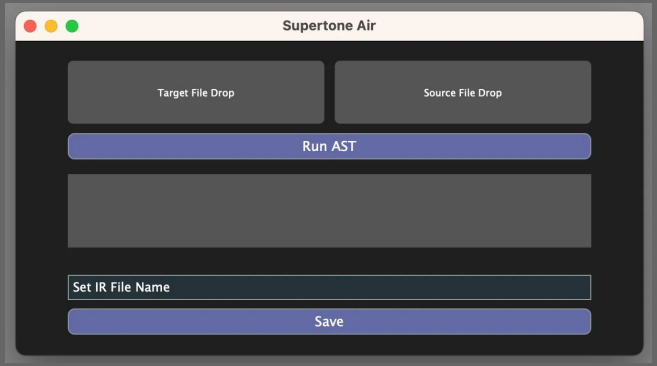






Developing Supertone Air

Development timeline



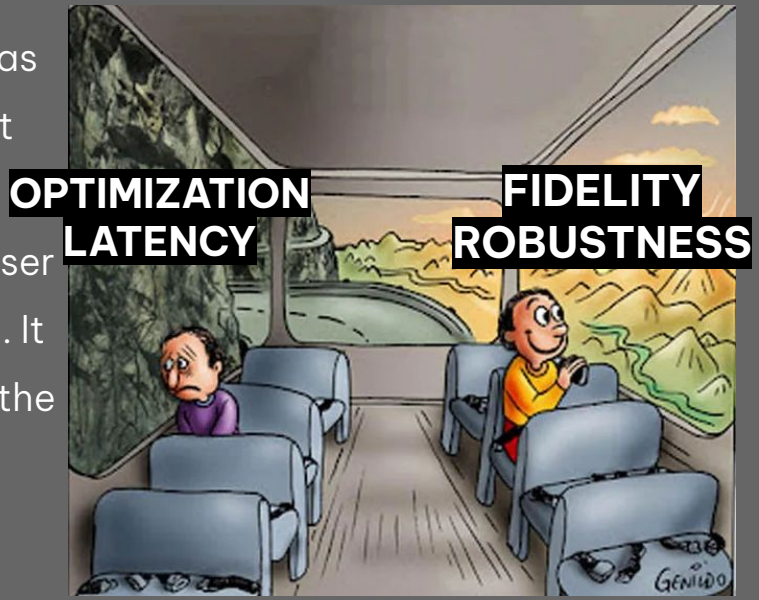
- Took less than 6 month including Market research, Product Design, Development.
- During the development, we took an advice from actual sound engineer to enhance the performance.

Developing Supertone Air

Different perspective

Performance!

- The new trained model has poorer performance.(Got slower)
- Let's just place the denoiser block before model input. It just makes simple.(To fit the deadline.)

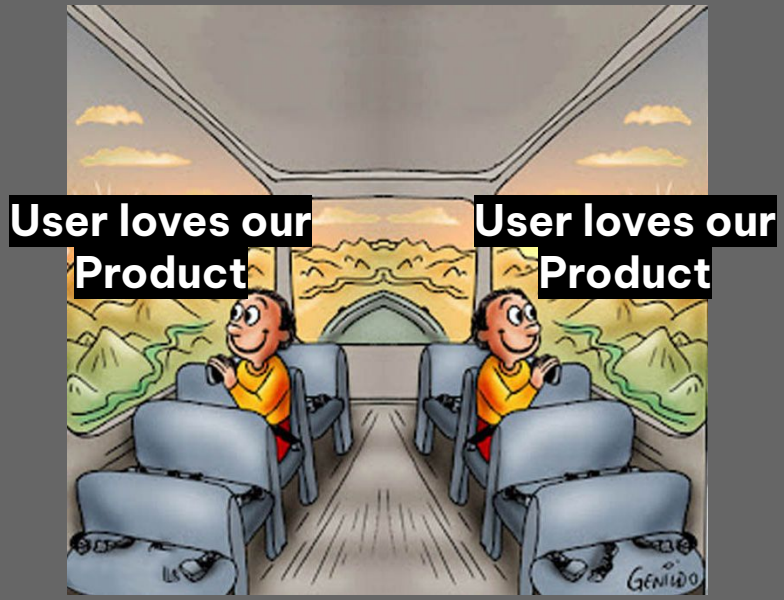


- It can't be! I've trained with more qualified data!(Got better fidelity)
- No. If the model has been well trained, than should handle input noise well. (which is not yet.)

Developing Supertone Air

Different perspective but same direction

Success!



Conclusions

Conclusions

- Build a good team.
  - Synergise you and your team’s expertise.
  - Effective communication is always gold.
- Focusing on User, not Technology itself.
  - Look at the problems to be solved, the needs to be met.
  - Solution depends on who our customers are.
- Have Fun!

SUPERTONE™ BEYOND THE VOICE

Thanks

Does anyone have any questions?

Email

rickysung@supertone.ai

Website

supertone.ai