Table of contents

Chapter 1

Introduction

1.1 Summary links

- **Contributor:** Michael Clemens
- **Mentors:** Jack Armitage, Chris Kiefer

1.2 Status

This project is currently just a proposal.

1.3 Proposal

1.4 About

- **Forum:** u/mclemcrew (Michael Clemens)
- **OpenBeagle:** mclem (Michael Clemens)
- **Github:** mclemcrew (Michael Clemens)
- **School:** University of Utah
- **Country:** United States
- **Primary language:** English
- **Typical work hours:** 7AM-3PM US Mountain
- **Previous GSoC participation:** N/A

Chapter 2

Project

Project name: Differential Logic Based Adaptive Soundwalk

2.1 Soundwalking

Soundwalking is a practice that was initially developed by the World Soundscape Project, an international research group founded by R. Murray Schafer (Traux 2021). The project focused on acoustic ecology and the idea that noise pollution in our sonic environment has impaired our ability to actively listen (Staśko-Mazur 2015). Soundwalking was initially used to improve our listening skills and understand the human experience of aural perception in modern society. Today, it is used to help listeners appreciate the importance of the soundscape in their environment and is widely used in soundscape research.

Other definitions of soundwalking from researchers include:

- An empirical method for identifying and analyzing the soundscape and its components in different locations (Adams, 2008)

- '...any excursion whose main purpose is listening to the environment' (Drever, 2009)

- A practice that promotes an aural perception of the environment as both a physical space and a space of social and political tensions, divisions, and flows (Carras, 2019)

In summary, soundwalking focuses primarily on the soundscape's auditory elements rather than other sources of sensory information. It is a way of connecting with nature that is often overlooked due to our reliance on visual cues. This project draws inspiration from the following innovative soundwalk concepts:

- **"Alter Bahnhof Video Walk"** by Janet Cardiff and George Bures Miller offers an uncanny soundwalking experience for users. Participants use an iPod and headphones to navigate Kassel's train station. The video on the screen mirrors their physical environment, and they are provided with a directional narration via the video provided.

- **"Ambulation"** by Shaw and Bowers focuses on in situ soundwalks. Sounds are collected and processed live from the participant's environment and broadcasted through wireless headphones, avoiding pre-recorded elements. Shaw initially implemented this on a laptop with PureData but now uses Bela as a way to realize this work.

This project proposal is a derivative of these works and leverages soundwalks as a medium for creative exploration. Users will wear wireless headphones while walking along a predetermined route provided by the aural narrator (similar to Cardiff and Miller's presentation). The system, using Bela, will continuously analyze the environment. It will identify specific audio patterns (e.g., footsteps on concrete, birdsong) and trigger curated narration tailored to the user's surroundings.

To ensure accurate, real-time sound classification, this project proposes implementing DiffLogic to address the latency issues commonly associated with deploying such models on embedded platforms such as Bela. The following section delves into the specifics of DiffLogic implementation.

2.2   Differential Logics

Traditional logic gate networks cannot conventionally leverage gradient-based training because of their non-differentiable nature. Petersen et al. (2022) address this limitation by introducing a relaxation process, making the gates differentiable and enabling standard training methods. Here is how their approach compares to classic neural networks:

A standard neural network uses a perceptron, which is a binary linear classifier and consists of the following four parts:

1. Input values or biases - A vector from a previous layer or discretized raw input (e.g., image pixel values, audio amplitudes)

2. Weights - Learned parameters demonstrating the importance of a particular node.

3. Weighted Sum - Inputs multiplied by weights, then summed

4. Activation Function - Non-linear function mapping the weighted sum to an output, enabling complex decision-making

The following image illustrates the steps of the perceptron.
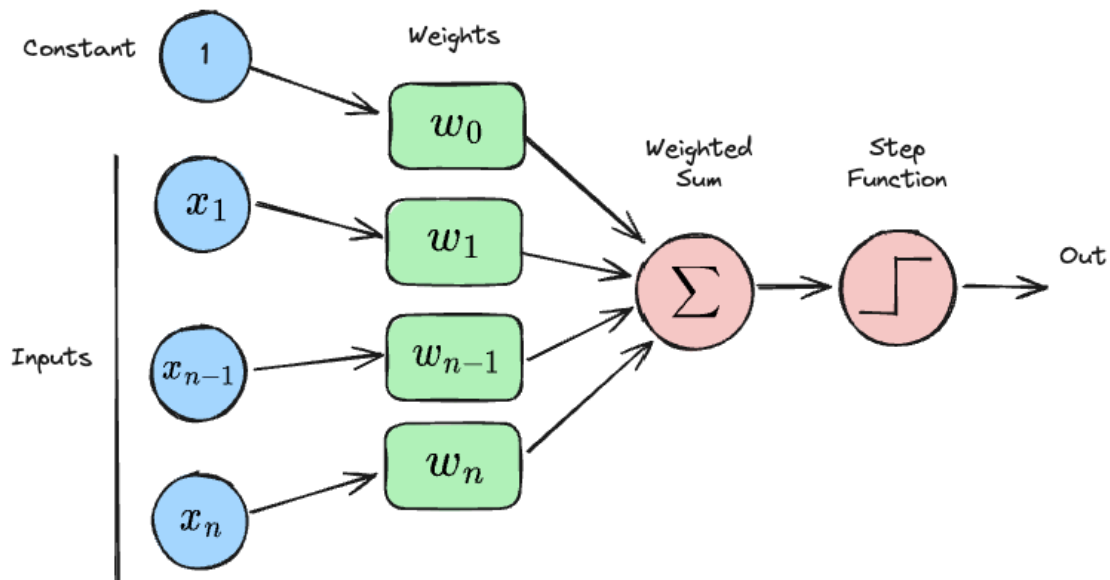
A logic gate is similar to a perceptron in that it also can be used as a binary linear classifier. With XOR, it takes on the implementation of a multi-layer perceptron, but the main concept within leveraging logic gates as perceptrons is that they both function as binary linear classifiers as demonstrated below:

AND

Inclusive OR

Exclusive OR

Inputs		Output
A	B	z
0	0	0
0	1	0
1	0	0
1	1	1

Inputs		Output
A	B	z
0	0	0
0	1	1
1	0	1
1	1	1

Inputs		Output
A	B	z
0	0	0
0	1	1
1	0	1
1	1	0
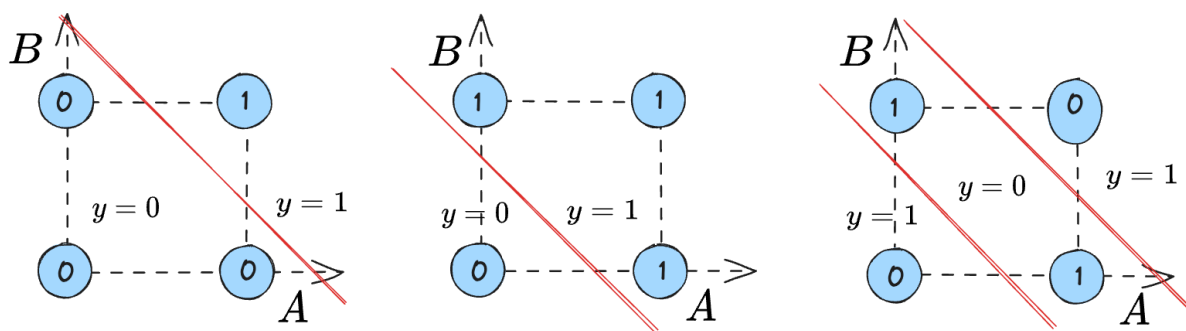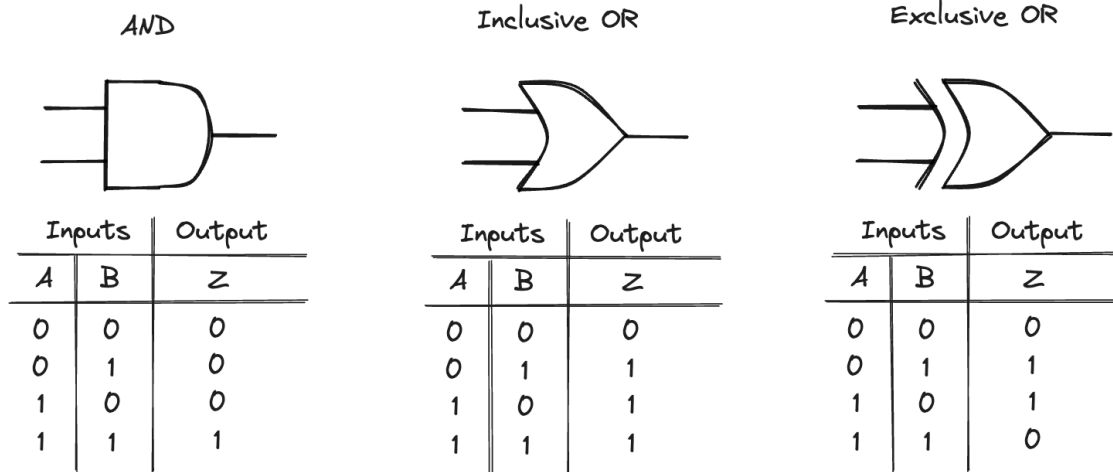
A table of key differences between the network architectures follows:

Logic Gates	Neurons
Can have non-linear decision boundaries (e.g., XOR)	Only models linear decision boundaries
Algorithm-like: Fixed function	Neural-like: Flexible learning of function
Fast inference	Slow inference

Although all logic gates can be modeled by using just the NAND and NOR gates (DeMorgan's Law), the authors of this work use the following 16, including a falsy and truthy representation as represented below:

ID	Operator	Real-valued	00	01	10	11
1	False	0	0	0	0	0
2	A ∧ B	A · B	0	0	0	1
3	¬(A ⇒ B)	A - AB	0	0	1	0
4	A	A	0	0	1	1
5	¬(A ⇐ B)	B - AB	0	1	0	0
6	B	B	0	1	0	1
7	A ⊕ B	A + B - 2AB	0	1	1	0
8	A ∨ B	A + B - AB	0	1	1	1
9	¬(A ≡ B)	1 - (A + B - 2AB)	1	0	0	1
10	¬B	1 - B	1	0	1	0
11	A ⇐ B	1 - B + AB	1	0	1	1
12	¬A	1 - A	1	1	0	0
13	A ⇒ B	1 - A + AB	1	1	0	1
14	¬(A ∧ B)	1 - AB	1	1	1	0
15	True	1	1	1	1	1

Leveraging these gates, the authors used gradient-based training to learn the optimal logic gate implementation for each node. Their classification models show that inference time is dramatically reduced while training costs are often higher. This speed advantage is crucial in applications like embedded computing, where models are typically small, and training occurs offline, making training cost less of a concern.

2.3   Related Works

Digital musical instruments (DMIs) demand low latency systems to ensure artists maintain full agency over their creative output. McPherson et al. (2016) underscore this need, demonstrating that common microcontrollers fail to meet accepted latency guidelines (generally between 4ms-10ms as supported by Lago and Kon (2004) and Wessel and Wright (2002)). Bela stands out in this context, offering submillisecond latencies and jitter under 50μs.

Bela's minimal delay makes it ideal for complex, interactive DMIs. As machine learning becomes increasingly prevalent in music and musical performances, creators require reliable ways to deploy these models on embedded AI systems. This was a key theme of the 2022 Embedded AI workshop at NIME led by this project's mentors and other researchers within this field (Pelinski et al. 2022). A previous GSoC project contributor (Pierce et al. 2023) presented at the workshop on the topic of achieving suitable performance within an embedded context using Bela. A follow-up to this work also came from (Pelinski et al. 2023) in their work *"Pipeline for recording datasets and running neural networks on the Bela embedded hardware platform."* These works collectively highlight efforts to streamline ML/DL deployment on the Bela platform.

Building upon this foundation, our mentors successfully trained a differential logic gate network on Bela for sound classification. Previous work by Solomes (2020) and Stowell demonstrates similar success using Bela for the DCASE 2018 - Bird Audio Detection task.

This proposal aims to extend these initiatives, explicitly targeting the intersection of NIME and embedded AI on Bela. The goal is to help empower musicians and makers to seamlessly create and utilize custom DMIs that harness the power of machine learning models.

2.4   Project Goals

There are two main goals of this project being as follows:

1. **DiffLogic Implementation Pipeline for Bela:**

   - **Comprehensive Guide:** Step-by-step instructions on building, training, and deploying deep logic gate models on Bela. This will include documentation, Jupyter notebooks, and video tutorials.

   - **Performance Metrics:** Evaluate the ideal use cases for this model architecture (e.g., sound classification, gesture recognition) and provide relevant performance metrics.

   - **Learning Resources:** Create supplementary materials to enhance understanding of DiffLogic.

2. **Soundwalk with Adaptive Narrative:**

   - Implement a soundwalk where the aural narrative dynamically updates based on fast, accurate sound classification of the user's sonic environment.

   - **Feature Exploration:** Utilize FFTs, GFCCs, and MFCCs for sound classification, potentially leveraging insights from Zhao (2013) on GFCCs' potential for accuracy improvement.

The majority of the project's effort will be dedicated to establishing a robust and user-friendly DiffLogic pipeline for Bela users. Initial focus will be on sound classification using the following datasets:

- UrbanSound8K: To evaluate the soundwalk implementation's effectiveness in a complex urban environment.

- DCASE 2018 - Bird Audio Detection: To facilitate direct performance comparisons between DiffLogic and existing CNN architectures on Bela.

If time permits, I will explore additional audio classification tasks such as speech recognition or sentence utterances. While gesture recognition could be highly valuable for NIME, it is currently considered outside the scope of this project.

2.5   Metrics

Building on the findings of Petersen et al. (2022), this project aims to achieve significant reductions in inference time without sacrificing accuracy when deploying models on Bela. We'll measure and compare the following metrics across classification tasks using the UrbanSound8K and DCASE 2018 - Bird Audio Detection datasets:

- Accuracy

- Training time

- Number of parameters

- Inference time

- Storage space

Models evaluated will include a decision tree learner, a logistic regression, a neural network, and a DiffLogic network.

The project's soundwalk experience will take place in the Indiana Dunes National Park in the latter half of summer 2024, featuring aural narrations by the project lead. We will fine-tune the classifier on a corpus of 500 labeled soundbites collected from the region. Participants will be provided a short explanation of what they will experience, will be guided to the origin, and the aural narrative will be provided in real time and will update accordingly based on their current aural surroundings.

2.6   Software

- Python (Jupyter, PyTorch)

- C

- C++

2.7   Hardware

- Bela

- Microphone

- BT Transmitter

2.8   Early Experiements / Troubleshooting

Within the difflogic repo in the main.py and main_baseline.py files, the Iris dataset is included in the code but not within the terminal commands. I updated their code and began attempting to run the experiments to see if I could verify and replicate their results. Unfortunately, I could not do so despite several attempts to get it working. Within my University's Center for High-Performance Computing, I ran into a GCC compiler version error (PyTorch required 9 and our node was running on 8). I tried to install it from source but ended up with more errors than anything and decided to spin up an EC2 instance instead and install everything manually to ensure the correct versioning. For CUDA to be uploaded, I needed to request that my vCPUs be upgraded in the region I was deploying so I could attach a GPU to my instance. After attempting to spin everything up from scratch, I kept receiving the following error while installing via pip or from building from source:

```
CUDA_HOME environment variable is not set. Please set it to your CUDA↵
↪install root.
```

After looking into this, I believe there is still a mismatch between torch versioning and CUDA, but I ran out of time to resolve this before getting anything running. The EC2 instance was the best solution for the time being, and despite following their installation support file, I still encountered issues I needed to resolve. I'll be looking through the CUDA installation guide in more detail since I'm sure I overlooked something while installing everything on the EC2 instance.

Chapter 3

Timeline

Provide a development timeline with 10 milestones, one for each week of development without an evaluation, and any pre-work. (A realistic, measurable timeline is critical to our selection process.)

Note: This timeline is based on the official GSoC timeline

3.1 Timeline summary

Date	Activity
February 26	Connect with possible mentors and request review on first draft
March 4	Complete prerequisites, verify value to community and request review on second draft
March 11	Finalized timeline and request review on final draft
March 21	Submit application
May 1	Start bonding
May 27	Start coding and introductory video
June 3	Release introductory video and complete milestone #1
June 10	Complete milestone #2
June 17	Complete milestone #3
June 24	Complete milestone #4
July 1	Complete milestone #5
July 8	Submit midterm evaluations
July 15	Complete milestone #6
July 22	Complete milestone #7
July 29	Complete milestone #8
August 5	Complete milestone #9
August 12	Complete milestone #10
August 19	Submit final project video, submit final work to GSoC site and complete final mentor evaluation

3.2 Timeline detailed

3.2.1 Community Bonding Period (May 1st - May 26th)

GSoC contributors get to know mentors, read documentation, get up to speed to begin working on their projects

3.2.2 Coding begins (May 27th)

Project Foundations

- Set up development environment for Bela platform

- Begin experimenting with pipeline for building ML/DL models and delopying/testing with Bela

3.2.3   Milestone #1, Introductory YouTube video (June 3rd)

- Begin implementing the pipeline for DiffLogic on Bela platform
- Experiment with different logic gate architectures and training methods
- Start drafting comprehensive guide and documentation for DiffLogic implementation

3.2.4   Milestone #2 (June 10th)

- Iterate on DiffLogic implementation based on initial testing and feedback
- Evaluate and select appropriate datasets for classifier training (UrbanSound8K, DCASE 2018)

3.2.5   Milestone #3 (June 17th)

- Begin training the DiffLogic model using selected datasets
- Evaluate performance metrics including accuracy, training time, and inference time
- Compare DiffLogic network with traditional classification models (decision tree, logistic regression, neural network)

3.2.6   Milestone #4 (June 24th)

- Develop supplementary materials such as Jupyter notebooks and video tutorials
- Complete writing comprehensive guide and documentation for DiffLogic implementation

3.2.7   Milestone #5 (July 1st)

- Develop supplementary materials such as Jupyter notebooks and video tutorials
- Start building soundwalking prototype
- Conduct initial tests and simulations to validate the effectiveness of real-time narrative updates based on sound classification

3.2.8   Submit midterm evaluations (July 8th)

- Implementation and results from DiffLogic implementation

Important: July 12 - 18:00 UTC: Midterm evaluation deadline (standard coding period)

3.2.9   Milestone #6 (July 15th)

- Start designing the soundwalk experience, including route planning and narrative structure
- Explore potential audio classification features (FFT, GFCCs, MFCCs) for sound classification
- Test initial sound samples with labels

3.2.10   Milestone #7 (July 22nd)

- Build a prototype of the soundwalk system integrating the DiffLogic classifier
- Conduct initial tests and simulations to validate the effectiveness of real-time narrative updates based on sound classification

3.2.11   Milestone #8 (July 29th)

- Optimize the soundwalk system for performance and efficiency
- Fine-tune the DiffLogic model based on feedback and performance evaluation
- Address any remaining issues or bugs in the implementation

3.2.12   Milestone #9 (Aug 5th)

- Finalize all project deliverables including documentation, tutorials, and the soundwalk system
- Conduct thorough testing and quality assurance checks to ensure everything functions as intended
- Prepare for the upcoming soundwalk experience in Indiana Dunes National Park

3.2.13   Milestone #10 (Aug 12th)

- Conduct the soundwalk experience in Indiana Dunes National Park
- Analyze feedback and data collected during the soundwalk experience
- Finalize deliverables

3.2.14   Final YouTube video (Aug 19th)

- Video of DiffLogic implementation and soundwalking example

Submit final project video, submit final work to GSoC site and complete final mentor evaluation

3.2.15   Final Submission (Aug 24nd)

Important: August 19 - 26 - 18:00 UTC: Final week: GSoC contributors submit their final work product and their final mentor evaluation (standard coding period)

August 26 - September 2 - 18:00 UTC: Mentors submit final GSoC contributor evaluations (standard coding period)

3.2.16   Initial results (September 3)

Important: September 3 - November 4: GSoC contributors with extended timelines continue coding

November 4 - 18:00 UTC: Final date for all GSoC contributors to submit their final work product and final evaluation

November 11 - 18:00 UTC: Final date for mentors to submit evaluations for GSoC contributor projects with extended deadline

Chapter 4

Experience and approach

Although I have limited experience with Bela specifically, I have worked on numerous embedded projects that demonstrate my proficiency in this area including my MIDI Mosaic project that utilized the Teensy microcontroller for processing of the capacitive touch sensor and MIDI data wrangling.

I'm currently building a co-creative music mixing agent that leverages non-copyrighted audio data to recommend audio effect, parameters, and their associated values based on user input through a chat-interface. Despite the potential challenges of using chat interfaces for music, as highlighted by Martin Mull's quote "Writing about music is like dancing about architecture." With this in mind, I'm leveraging XAI principles such as chain-of-thought reasoning and counterfactuals to help recommend mixing basics for amateurs through pro-ams. This project demonstrates my experience in applying advanced machine learning techniques to real-world problems.

Through my research within the field of natural language processing (NLP) and my hands-on experience with various projects, I have developed a strong foundation in machine learning and deep learning. While I am aware of some Diff Logic approaches from papers such as "High-Fidelity Noise Reduction with Differentiable Signal Processing", I have not yet had the opportunity to implement them directly. However, I am excited to apply my existing knowledge and skills to explore this concept further within the context of embedded AI.

4.1   Contingency

There are a few places to reach out for help, including:

- The Beagle Forum or Discord

- The Bela Forum

- X (Formerly Twitter)

- DASP-PyTorch

- Diff Logic PyTorch library

- Bela DL Pipeline

- C++ Real-time Audio Programming with Bela

- PRU Programming for Beagleboard

- Compiled models on RPi2040

- Experiments with DiffLogic and Bela

- PyBela

Each of these resources provides benefit to the overall project in some manner. The Beagle Forum or Bela Forum are the optimal places to ask for assistance for Bela-related queries. Within the differential logic and ML/DL communities, X has been helpful with connecting to the right body of work or the right person who may

be able to help briefly. Differential signal processing has become big recently within Music Generation, and I would reach out to researchers I know within these projects should I require assistance before I meet with my mentor again.

4.2   Benefit

If completed, this project will have a positive impact on the BeagleBoard community by providing a robust and accessible framework for implementing embedded AI designs, specifically in the domain of machine listening and audio processing.

The project will demonstrate the feasibility and effectiveness of leveraging Differential Logic techniques for real-time audio processing on the BeagleBoard platform. By showcasing the successful integration of advanced machine learning algorithms, such as Siamese networks and transfer learning, with the BeagleBoard's hardware capabilities, this project seeks to inspire and encourage other developers and researchers to explore similar approaches in their own work.

This project will also contribute a set of demos, code snippets, and tutorials for implementing on-device machine learning models, sound classification, and machine listening using differential logic on the Bela platform. These resources will serve as a foundation for the BeagleBoard community to build upon, ensuring that embedded AI projects are more accessible to the overall community within BB and NIME.

Relevant Forum Posts:

- Embedded differentiable logic gate networks for real-time interactive and creative applications
- Enhanced Media Experience with AI-Powered Commercial Detection and Replacement

4.3   Misc

This is the link to the merge request.

4.4   References

- X. Zhao and D. Wang, "Analyzing noise robustness of MFCC and GFCC features in speaker identification," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, 2013, pp. 7204-7208, doi: 10.1109/ICASSP.2013.6639061.

- Adams, Bruce, N. C., Davies, W., Cain, R., Jennings, P., Carlyle, A., Cusack, P. T., Hume, K., & Plack, C. (2008). *Soundwalking as a methodology for understanding soundscapes*. https://www.semanticscholar.org/paper/Soundwalking-as-a-methodology-for-understanding-Adams-Bruce/f9e9eb719fb213c3f1f72dd000cabcf52169e5c7

- Carras, C. (2019). Soundwalks: An experiential path to new sonic art. *Organised Sound*, *24* (3), 261–273. https://doi.org/10.1017/S1355771819000335

- Drever, J. L. (2009). *Soundwalking: Aural Excursions into the Everyday*. https://www.semanticscholar.org/paper/Soundwalking%3A-Aural-Excursions-into-the-Everyday-Drever/870b40f70636933df0efb2d9706dbea678f3cf3a

- Staśko-Mazur, K. (2015). Soundwalk as a multifaceted practice. *Argument: Biannual Philosophical Journal*. https://www.semanticscholar.org/paper/Soundwalk-as-a-multifaceted-practice-Sta%C5%9Bko-Mazur/571241ba10d121a0b6d8e29cc7efb73387eaa436

- Truax, B. (2021). R. Murray Schafer (1933–2021) and the World Soundscape Project. *Organised Sound*, *26* (3), 419–421. https://doi.org/10.1017/S1355771821000509

- Pierce, E., Shepardson, V., Armitage, J., & Magnusson, T. (2023). Bela-IREE: An Approach to Embedded Machine Learning for Real-Time Music Interaction. AIMC 2023. Retrieved from https://aimc2023.pubpub.org/pub/t2l10z49

- Pelinski, T., Shepardson, V., Symons, S., Caspe, F. S., Benito Temprano, A. L., Armitage, J., ... McPherson, A. (2022). Embedded AI for NIME: Challenges and Opportunities. International Conference on New Interfaces for Musical Expression. https://doi.org/10.21428/92fbeb44.76beab02

- Solomes, A. M., and D. Stowell. 2020. "Efficient Bird Sound Detection on the Bela Embedded System." https://qmro.qmul.ac.uk/xmlui/handle/123456789/67750.

- Shaw, Tim, and J. Bowers. 2020. "Ambulation: Exploring Listening Technologies for an Extended Sound Walking Practice." *New Interfaces for Musical Expression*, 23–28.

- McPherson, Andrew P., Robert H. Jack, Giulio Moro, and Others. 2016. "Action-Sound Latency: Are Our Tools Fast Enough?" *Qmro.qmul.ac.uk*. https://qmro.qmul.ac.uk/xmlui/handle/123456789/12479.

- Pelinski, Teresa, Rodrigo Diaz, A. L. B. Temprano, and Andrew J. McPherson. 2023. "Pipeline for Recording Datasets and Running Neural Networks on the Bela Embedded Hardware Platform." *ArXiv* abs/2306.11389 (June). https://doi.org/10.48550/arXiv.2306.11389.

- N.P. Lago and F. Kon. The quest for low latency. In Proc. ICMC, 2004.

- D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, *26*(3):11–22, 2002.