Project X Report: Sound Scout

Samuel David Kleiner

August 2024

1 Introduction

This project evolved from its initial proposal as I immersed myself in the field of Sound Source Localization [SSL] with Deep Learning, and focused on Sound Event Localization and Detection [SELD]. My passion for my research project comes from my family history. My sister is half deaf, so she cannot locate the origin of sounds. This is a problem known as spatial hearing loss. Spatial hearing loss is a greater problem than it might initially appear to people with full hearing. Children with spatial hearing loss have difficulty understanding speech in the classroom leading to worse learning outcomes since humans use the "spatial distribution of sources to suppress unwanted signals" [3]. Spatial hearing is crucial for humans' natural ability to filter out background noise and focus on one source of audio. When we focus on a teacher's voice we subconsciously use spatial hearing to help us ignore the noise around us. One study, which confirmed the results of several earlier studies, "demonstrated a 12-dB improvement in performance, referred to as spatial advantage. [...] participants were able to understand the target sentences at a 12-dB worse signal-to-noise ratio (SNR) once the signal and the noise were separated and they could use spatial processing to assist them with the task" [4].

After reading [5], I learned of the DCASE2024 challenge and focused on Task 3: Audio and Audiovisual Sound Event Localization and Detection with Source Distance Estimation, as a useful conduit for my research. I chose to focus on the First-Order Ambisonics [FOA] format, and a multi-ACCDOA approach, the benefits of which are described in [8].

The final goal was to reproduce the results of the baseline SELD system, and reduce training time, so that future improvements to the system could be trained and tested faster. My results from reproducing the baseline and testing different batch sizes and correlated learning rates, are shown in 1 and 2. The baseline results were successfully reproduced, and the batch sizes were not found to have a significant negative effect on testing accuracy as described in [6], but the batch sizes did not have a significant effect on the training time. A batch size of 512 with GPU (Trial 3a) did have the lowest training time out of all trials using a GPU, and it was less than half of the time of Trial 3b, which did not use a GPU. This was fascinating since GPU use was very low, only 3% on average

for all GPU trials. This result suggested that loading the data onto the GPU took the vast majority of the CPU compute time, and that the GPU was critical for a few computationally intensive parts of the program. To better understand this phenomenon, I profiled the GPU usage using line_profiler. It turned out that the data transfer did not take up a high percentage of the job's time. It was actually the determine_similar_location function that consumed the majority of the compute time. The test_epoch function consumed 86.6% of the total compute time. Within, the test_epoch function, the determine_similar_location function consumed 69.3% of the compute time. I tried to optimize the determine_similar_location function to reduce the total compute time of training and testing this SELD system, but I was not successful in the time I had. I learned so much during this project, and I will probably continue to look into this on my own.

2 Reproducing and Accelerating the Baseline

The first step in reproducing the baseline system was learning how Princeton's High Performance computing cluster worked. I worked through the tutorial, Getting Started with the Research Computing Clusters Video Version. Using the tutorial's lessons, I set up a conda environment to test the baseline code in, and wrote a slurm job file. I successfully extracted the labels from the data. Then I did small tests with the training program using only a subset of the data and a limited number of epochs. When that was successful I gradually requested more resources from Adroit until the training on the full dataset with 250 epochs was successful. My results show the baseline was reproduced using code from the DCASE 2024 SELD Repository with "Published" being the baseline's published results, and "Reproduced" being my own. The published results used F20, a more limited version of the F-Score metric, which is a measure of how accurate the classification of the sounds are, and did not include a SELD score or distance metric.

Test Parameters	**Reproduced**	**002**	**003a**	**003b**	**004**
Batch Size	128	256	512	512	1028
Learning Rate	0.001	0.002	0.004	0.004	0.008
GPUs	1	1	1	0	1
CPU Cores	1	1	1	1	1
Run Time	08:15:35	08:22:16	07:38:50	18:23:28	08:26:46
CPU Utilization (%)	99	99	100	100	100
CPU Memory Usage (%)	25	28	68	63	61
GPU Utilization (%)	3	2	3	NA	2
GPU Memory Usage (%)	10	18	35	NA	68

Table 1: Comparison of Adroit Resources Used Across Trials

Metric	Published	Reproduced	002	003a	003b	004
SELD Score	NA	0.52	0.49	0.56	0.49	0.61
SED F-score	13.1	15.8	14.1	13.8	14.1	13.2
DOA Angular Error	36.9	37.4	37.6	33.2	38.5	33.7
Distance Metrics	NA	0.53	0.54	0.50	0.54	0.55
Relative Distance Metrics	0.33	0.29	0.29	0.29	0.28	0.30

Table 2: Comparison of SELD Metrics Across Trials

3 Lessons Learned

The tutorial by Princeton Research Computing was critical to successfully writing a slurm script, setting up a conda environment on Adroit, transferring code and data to Adroit, and running my jobs. I learned how to navigate Adroit and Princeton's High Computing Resources to test programs and allocate resources properly. One useful rule in Princeton Research Computing's introduction to using their computing clusters is to maintain an organized file system with a README, so that in the event someone else ever has to continue your research or look at your work, it is as easy as possible to do so. I set up my initial file system on Adroit with the goal of maintaining an organized system, and copied the tutorial's suggested file system of separating files, data, and projects. Yet, even with my preparation in writing this report, I realized I had named my output files with names that were not significantly clear as to what trial went with what job_id, which mattered in my case since I was also testing different parameters affect on training time and CPU/GPU utilization. I was able to figure out which job went with which output file, but it was a good reminder that organization in your file system and data must be rigorously maintained.

While learning the ins and outs of using Princeton Research Computing for machine learning, I also learned about the deep field of SELD research. I learned the importance of data augmentation methods in a field where data collection tends to be expensive and time-intensive.

I would like to thank my advisor Professor Hossein Valavi for his guidance over the summer.

4 Material Expenses

With my original proposal, I submitted a budget that was mainly required for data collection. As my project evolved, I found data sets that worked for my research very well, and I was able to bypass doing data collection for this summer. Thus, I did not need to use any of the $480 approved for my project, but I appreciate the generosity of the SEAS undergraduate research program.

References

[1] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):34–48, Mar. 2019.

[2] D. Berghi, P. Wu, J. Zhao, W. Wang, and P. J. B. Jackson. Fusion of audio and visual embeddings for sound event localization and detection, 2023.

[3] D. H. Cameron S. The listening in spatialized noise-sentences test (lisn-s): comparison to the prototype lisn and results from children with either a suspected (central) auditory processing disorder or a confirmed language disorder. *Journal of the American Academy of Audiology*, 2008.

[4] H. Glyde, L. Hickson, S. Cameron, and H. Dillon. Problems hearing in noise in older adults: a review of spatial processing disorder. *Trends in Amplification*, 2011.

[5] P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin. A survey of sound source localization with deep learning methods. *The Journal of the Acoustical Society of America*, 152(1):107–151, July 2022.

[6] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima, 2017.

[7] D. A. Krause, A. Politis, and A. Mesaros. Sound event detection and localization with distance estimation, 2024.

[8] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji. Multi-ACCDOA: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, May 2022.

[9] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji. Multi-accdoa: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 316–320, 2022.

[10] P. A. Sudarsanam, A. Politis, and K. Drossos. Assessment of self-attention on learned features for sound event localization and detection. In *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, pages 100–104, November 2021.

[11] T. N. Tho Nguyen, D. L. Jones, K. N. Watcharasupat, H. Phan, and W.-S. Gan. Salsa-lite: A fast and effective feature for polyphonic sound event

localization and detection with microphone arrays. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 716–720, 2022.