# Alternative implementations of the Auxiliary Duplicating Permutation Invariant Training

Jens Gulin[†*], Kalle Åström[†]
[*] Sony Europe B.V., Sweden [†] Lund University, Sweden

## Motivation & Contribution | Faster ADPIT means better multi-source sound localization

Simultaneous **sound event localization and detection** (SELD) extracts the class and location of the sound sources from audio. To manage events where the **same class occurs at multiple locations** at the same time, Multi-ACCDOA[1] with Auxiliary Duplicating Permutation Invariant Training (ADPIT) is powerful. The baseline system for the DCASE SELD challenge 2024[2] has an implementation of ADPIT. In this paper **we clarify the permutation requirement** and discuss **alternative ways to implement ADPIT** to make the equivalent loss calculations faster. Since ADPIT scales poorly for an increased number of slots, improved efficiency is of general interest for audio localization. The results indicate a substantial **reduction of computation needed**, which opens for new possibilities.

## RESULTS | Indicating substantial speed-up

Proposals are not yet implemented. An estimated execution cost ($\mathcal{N}_{iter}$) is shown in Table 1, for a fictive dataset similar to STARSS23[3] with $13$ classes, $200\mathtt{k}$ frames, all $M \leq 5$. Only $30\mathtt{k}$ frames have $M > 1$, and even there it is unlikely that the overlapping events are of the same class.

**Table 1:** Approximative execution cost improvement gain factor relative to baseline. The pessimistic $\mathcal{N}_{scopes}$ value for BRADPIT adds the $M > 1$ category miscounted by the optimistic case. $\mathcal{N}_{pairs}$ for these scopes is the "dist. first" minus the optimistic number. `k` is thousands and `M` is millions.

| | $N = 3$ *(with frame-class scope)* | | | | $N = 5$ *(with frame scope)* | | | |
|---|---|---|---|---|---|---|---|---|
| *Permutation Strategy* | $\mathcal{N}_{scopes}$ | $\mathcal{N}_{pairs}$ | $\mathcal{N}_{iter}$ | *gain* ↑ | $\mathcal{N}_{scopes}$ | $\mathcal{N}_{pairs}$ | $\mathcal{N}_{iter}$ | *gain* ↑ |
| Baseline | $200\mathtt{k} \times 13$ | $13 \times 3$ | $101\mathtt{M}$ | $1$ | $200\mathtt{k}$ | $540 \times 5$ | $540\mathtt{M}$ | $1$ |
| ARMPIT opt. overlay | $200\mathtt{k} \times 13$ | $6 \times 3$ | $46\mathtt{M}$ | $2.16$ | $200\mathtt{k}$ | $240 \times 5$ | $240\mathtt{M}$ | $2.25$ |
| ARMPIT dist. first | $200\mathtt{k} \times 13$ | $10$ | $26\mathtt{M}$ | $3.9$ | $200\mathtt{k}$ | $< 100$ | $< 20\mathtt{M}$ | $> 27$ |
| BRADPIT (optimistic) | $200\mathtt{k} \times 13$ | $1 \times 3$ | $7.8\mathtt{M}$ | $13$ | $200\mathtt{k}$ | $1 \times 5$ | $1\mathtt{M}$ | $540$ |
| BRADPIT (pessimistic) | $+30\mathtt{k}$ | $10 - 3$ | $8.0\mathtt{M}$ | $12.7$ | $+30\mathtt{k}$ | $100 - 5$ | $3.85\mathtt{M}$ | $140$ |

Basic cost setup:

$$\mathcal{N}_{scopes} = \mathcal{N}_{frames} \times \mathcal{N}_{cls}$$
$$\mathcal{N}_{pairs} = \mathcal{N}_{seqs} \times N$$
$$\mathcal{N}_{iter} = \mathcal{N}_{scopes} \times \mathcal{N}_{pairs}$$

Different implementation strategies alter the setup. Note: $\mathcal{N}_{pairs}$ may be different in each scope.

## Proposals | ARMPIT & BRADPIT

The premise is that a fixed number $N$ **output slots per scope** are filled with SELD-token predictions. The **true number of events,** $M$, for a particular scope is known from Ground Truth (as well as the class and location). ADPIT calculates the loss for each permutation and keeps the minimum one. The Baseline processes each permutation separately.

We propose a number of simple and elegant implementation strategies:
**ARMPIT** (ADPIT with Reduced Multiplications): Overlays the M-categories to avoid useless calculations. Finds the loss (distance) of each pair separately, then calculates the permutation sums.
**BRADPIT** (Branch Remedied ADPIT): Branches to different branch-less implementations based on the maximum $M$ of the scope or batch. With mostly $M \leq 1$, the loss is often permutation free.

**Table 2:** The sequences of $N = 3$ ARMPIT already overlayed, using dummy event sums named X$j$. a) The dummy sequences with the events in slot order, and one permutation (based on category) per row. The token-event pairs enumerated ① to ⑫. b) The unique dummy event sums. As before, only A, B *or* C is non-zero, depending on $M$.

| *a) Dummy sequences* | |
|---|---|
| X1 X2 X3 | ① ② ③ |
| X1 X3 X2 | ① ④ ⑤ |
| X0 X1 X3 | ⑥ ⑦ ③ |
| X0 X3 X1 | ⑥ ④ ⑧ |
| **X4 X5 X0** | **⑫ ⑪ ⑩** |
| X3 X2 X1 | ⑨ ② ⑧ |

| *b) Dummy event sums* |
|---|
| X1 = C0+B0+A0 |
| X2 = C1+B0+A0 |
| X3 = C2+B1+A0 |
| **X4 = C2+B0+A0** |
| **X5 = C0+B1+A0** |
| X0 = C1+B1+A0 |

Pairs ⑫ ⑪ ⑩ and sums X4 and X5 are not reused. X4X5X0 can be replaced with two sequences, X3X1X2 (⑨ ⑦ ⑤) and X1X3X0 (① ④ ⑩). This makes ten pairs in seven dummy sequences and four dummy sums to find the min loss. The minimal number of pairs for $N = 5$ has not yet been fully investigated and is estimated to be $< 100$.

## Permutations | A combinatorial explosion

We do not change the $\mathcal{O}$ time complexity and the asymptotic cost of $N$. However, with $N$ fixed and small, efficiency is about avoiding a large constant factor, prohibiting to both large and small sample sizes.

### AUXILIARY DUPLICATING FOR ADPIT

ADPIT duplicates output when there are fewer events than slots. Mapping token-event pairs can be seen as placing $N$ distinct tokens in $M$ labeled piles, leaving no pile empty. The number of permutations[4, Count B3] is,

$$\underset{\text{for } 1 \leq M \leq N}{K(N, M)} = M!\, S(N, M) \tag{1}$$
$$= \sum_{i=0}^{M} (-1)^{M-i} \binom{M}{i} i^N = M! \sum_{i=0}^{M} \frac{(-1)^{M-i}\, i^N}{(M-i)!\, i!},$$

where $S(N, M)$ are known as the *Stirling numbers of the second kind*, defined as the number of ways to partition a set of $N$ elements into $M$ non-empty subsets. The On-Line Encyclopedia of Integer Sequences (OEIS) has (1) as entry A019538.

## References

[1] Multi-ACCDOA: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training
K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, et al. ICASSP 2022.

[2] GitHub: DCASE2024_seld_baseline
https://github.com/partha2409/DCASE2024_seld_baseline/tree/0b07887

[3] STARSS23: An Audio-Visual Dataset of Spatial Recordings of Real Scenes with Spatiotemporal Annotations of Sound Events
K. Shimada, A. Politis, P. Sudarsanam, D. A. Krause, et al. NeurIPS 2023.

[4] Let's Expand Rota's Twelvefold Way For Counting Partitions!
R. A. Proctor. ArXiv pre-print 2007, arXiv:math/0606404.

This work was partially supported by the strategic research project ELLIIT, and by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.