BEATS WITH CROSS-ATTENTION FOR MULTI-CHANNEL AUDIO ALIGNMENT

Technical Report

Ragib Amin Nihal, Benjamin Yen, Takeshi Ashizawa, Kazuhiro Nakadai

Institute of Science Tokyo
Department of Systems and Control Engineering
Tokyo, Japan

1. ABSTRACT

W modified the provided BEATs baseline with three main changes. First, we added cross-attention layers that allow audio embeddings from different channels to interact before making alignment predictions. Second, improved the training process with better data sampling, conservative augmentation (amplitude scaling and noise addition), and AdamW optimization with learning rate scheduling. Third, replaced the baseline's binary counting similarity metric with confidence-weighted scoring that uses the full range of model outputs. The system uses the same candidate generation approach as the baseline but processes alignment decisions differently. On validation data, the method achieved MSE of 0.099 for ARU and 0.521 for zebra finch.

Index Terms— audio synchronization, multi-channel alignment, cross-attention, BEATs, clock drift

2. INTRODUCTION

Multi-channel audio synchronization addresses the problem of temporal alignment between recordings from multiple devices that suffer from nonlinear clock drift. The BioDCASE 2025 Task 1 challenge provides ARU and zebra finch datasets with desynchronization up to ±5 seconds, where each audio file contains keypoints indicating corresponding timestamps between channels (Figure 1).

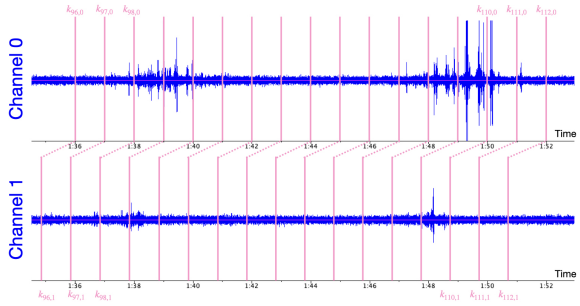

Figure 1: Keypoint-based alignment visualization showing desynchronized channels with corresponding timestamps (adapted from BioDCASE 2025 challenge description [2]).

The provided baseline employs a BEATs encoder [1] with binary classification to determine if audio clips are temporally aligned. This work extends the approach with architectural modifications and improved training procedures.

3. METHOD

3.1. Architecture

The system extends the baseline BEATs encoder architecture with cross-attention mechanisms that enable interaction between audio embeddings from different channels before making alignment predictions. Figure 2 shows the complete system architecture.

Cross-Attention Module: A multi-head cross-attention layer [3] is introduced (Figure 3) that processes concatenated embeddings from both audio channels:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q, K, and V are derived from the combined channel embeddings. This allows the model to capture temporal relationships between channels more effectively than simple concatenation.

Enhanced MLP Head: The baseline's single hidden layer MLP is replaced with a deeper network including:
- Increased capacity (256-128-64-1 architecture)
- Layer normalization and GELU activations
- Residual connections for training stability
- Dropout regularization (0.1-0.05 schedule)

3.2. Training Improvements

Data Sampling: Modified the baseline's random sampling by using every 20th keypoint per file, providing better diversity while maintaining temporal coverage.

Conservative Augmentation: Applied to 30% of samples:
- Amplitude scaling: ±10% variation
- Additive noise: SNR 40-50 dB
- Consistent augmentation across both channels

Optimization: Replaced Adam with AdamW (weight decay: 0.01) and ReduceLROnPlateau scheduling for more stable training convergence.

3.3. Confidence-Weighted Scoring

The key enhancement over the baseline lies in the scoring strategy. Rather than relying on a binary count of predictions (i.e., summing the number of outputs exceeding a threshold, $\sum(\text{pred} > 0)$), the system adopts a confidence-weighted scoring function that incorporates multiple aspects of the prediction distribution (see Figure 4, Table 1):

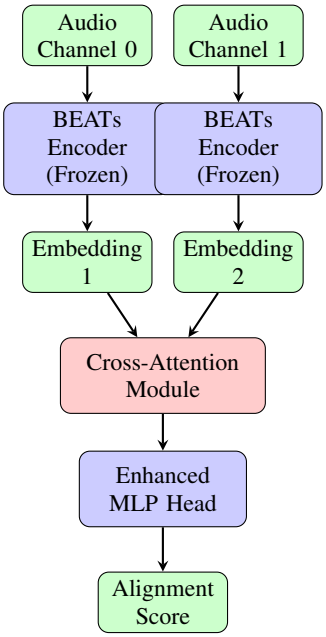

Figure 2: System Architecture Overview. Audio from both channels is processed through frozen BEATs encoders, then cross-attention allows interaction between embeddings before the enhanced MLP predicts alignment scores.

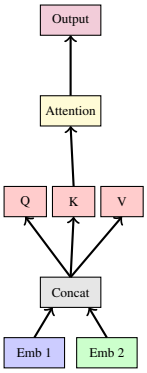

Figure 3: Cross-Attention Mechanism. Embeddings from both channels are concatenated and processed through multi-head attention to capture cross-channel temporal relationships.

$$\text{Score} = 0.4 \cdot \mu_{\text{pos}} \cdot r_{\text{pos}} + 0.3 \cdot \mu_{\text{top}} + 0.2 \cdot \sum_i \sigma(p_i) + 0.1 \cdot e^{\mu} \quad (2)$$

where:

- μ_{pos} is the mean confidence score among positively predicted instances,
- r_{pos} is the ratio of positive predictions to the total number of predictions,
- μ_{top} denotes the mean of the top 25% prediction confidences (i.e., the top quartile),
- $\sigma(p_i)$ is the sigmoid transformation applied to each prediction score p_i,
- μ is the global mean prediction score across all instances.

The final term, e^{μ}, emphasizes predictions with globally high confidence. The weights (0.4, 0.3, 0.2, 0.1) were empirically selected to balance the influence of precision, coverage, and confidence calibration.

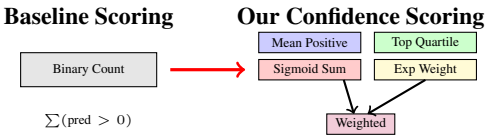

Figure 4: Scoring Mechanism Comparison. Our confidence-weighted approach leverages multiple prediction statistics versus the baseline's simple binary counting.

3.4. Inference Process

The inference follows the baseline's candidate generation approach but modifies the scoring mechanism (Figure 5):

1. Generate candidate keypoint sets assuming linear drift + constant offset
2. For each candidate set, extract 1-second audio clips at keypoint locations
3. Score each candidate using the enhanced model and confidence weighting
4. Select the candidate with the highest confidence-weighted score

Table 1 compares the scoring mechanisms in detail.

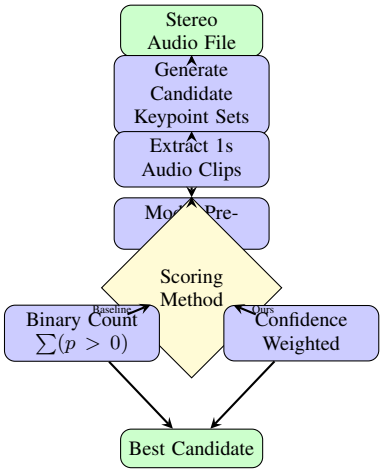

Figure 5: Inference process comparison. Both approaches use identical candidate generation but differ in how predictions are scored to select the best alignment.

Table 1: Scoring Mechanism Comparison

Aspect	**Baseline**	**Our Method**
Score Type	Binary count	Confidence weighted
Formula	$\sum(\text{pred} > 0)$	Weighted combination of 4 components
Confidence Use	Threshold only	Full prediction range
Information Loss	High (binary)	Low (continuous)

4. DATASET ANALYSIS

The challenge provides two datasets with different characteristics:

ARU Dataset: Contains 36 training files and 12 validation files from passive automated recording units. Based on analysis, ARU data exhibits higher drift variability.

Zebra Finch Dataset: Contains 108 training files and 16 validation files from zebra finch recordings. This dataset shows more consistent patterns .

Both datasets contain keypoints at 1-second intervals with drift constrained to ±5 seconds. The domain shift between datasets presents challenges for joint training approaches, as evidenced by different optimal model parameters for each domain.

5. EXPERIMENTAL SETUP

Datasets: Used the provided ARU and zebra finch train/validation splits without external data.

Training Configuration:
- Batch size: 32 (reduced to prevent $O(B)$ memory scaling)
- Learning rate: 2×10^{-4} with ReduceLROnPlateau scheduling
- Epochs: 100 with early stopping (patience: 25)
- BEATs encoder frozen during training
- Optimizer: AdamW with weight decay 0.01

Hardware: Training performed on NVIDIA H100 GPU with CUDA optimization and memory management (cache clearing every 5 batches to prevent fragmentation).

6. RESULTS

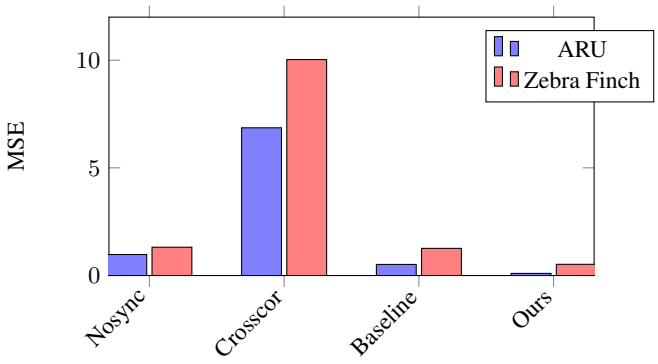

Figure 6: Validation MSE comparison across methods. Our approach achieves substantial improvements over all baselines on both datasets.

Table 2 shows detailed validation results compared to the provided baselines.

Table 2: Validation MSE Results

Method	ARU	Zebra Finch
Nosync	0.976	1.315
Crosscor	6.861	10.029
Deep learning baseline	0.516	1.262
Ours	**0.099**	**0.521**

Our method achieved MSE reductions of 80.8% for ARU data and 58.7% for zebra finch data compared to the deep learning baseline (Figure 6).

7. ANALYSIS

Component Contributions: Confidence-weighted scoring provided the largest performance improvement, followed by cross-attention mechanisms and training enhancements. The enhanced MLP architecture improved training stability with modest accuracy gains.

Dataset Differences: The method showed larger improvements on ARU data compared to zebra finch data. This may be attributed to ARU's higher drift variability providing more diverse training signals for the cross-attention mechanism to learn from.

Limitations: The approach still relies on the baseline's linear drift assumption for candidate generation, which may not capture all real-world drift patterns. The cross-attention mechanism adds computational overhead during training, though this is mitigated by freezing the BEATs encoder.

8. CONCLUSION

We presented modifications to the BEATs baseline for multi-channel audio alignment. The combination of cross-attention mech-

anisms, enhanced training procedures, and confidence-weighted scoring produced improved performance on both ARU and zebra finch datasets. The approach demonstrates that focused architectural modifications and better utilization of model outputs can improve alignment accuracy without requiring external data.

Future work could explore adaptive candidate generation based on learned drift patterns and investigation of the cross-attention mechanism's learned temporal relationships.

9. REFERENCES

[1] Microsoft Research. BEATs: Audio Pre-Training with Acoustic Tokenizers. *arXiv preprint arXiv:2212.09058*, 2022.

[2] Hoffman, B., Gill, L., Heath, B., Narula, G. BioDCASE 2025 Task 1: Multi-Channel Alignment. *BioDCASE Challenge*, 2025.

[3] Vaswani, A., et al. Attention is All You Need. *Advances in Neural Information Processing Systems*, 2017.

[4] Loshchilov, I., Hutter, F. Decoupled Weight Decay Regularization. *International Conference on Learning Representations*, 2019.

[5] Sakoe, H., Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43-49, 1978.

[6] Knapp, C., Carter, G. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320-327, 1976.