IMPROVING TEMPORAL BOUNDARY PRECISION IN AUDIO MOMENT RETRIEVAL

Technical Report

Ren Usui[1]*, Ryuta Fujimoto*[1]*, Mikuri Kikuchi*[1]*, Taichi Kitao*[1]*, Tomohisa Suzuki*[1]*,*

[1] Yokohama City University, Graduate School of Data Science
22-2, Seto, Kanazawa-ku, Yokohama, Kanagawa, Japan

ABSTRACT

This technical report describes our system for Audio Moment Retrieval (AMR) from long audio in the DCASE 2026 challenge (task6). To improve boundary localization and retrieval accuracy, we build our system on the Unified Video Comprehension framework (UVCOM) and introduce four main modifications: M2D-CLAP-based audio and text feature extraction, Varifocal Loss for IoU-aware confidence estimation, random query sampling during pretraining, and inference-time boundary refinement based on saliency scores. Experimental results on the CASTELLA test dataset show that the proposed modules consistently improve retrieval performance, with the submitted systems substantially outperforming the official QD-DETR baseline across all evaluation metrics. The best configuration achieves a Recall@0.7 of 40.14, demonstrating the effectiveness of the integrated approach for robust AMR in real-world long audio recordings.

Index Terms— Audio Moment Retrieval, UVCOM, M2D-CLAP, Varifocal Loss, Random Query Sampling, Boundary Refinement

1. INTRODUCTION

Audio Moment Retrieval (AMR) aims to identify temporal moments in long audio recordings that correspond to natural language queries. Unlike conventional clip-based retrieval, AMR requires an understanding of temporal context over several minutes and precise cross-modal alignment between text and audio.

Recent studies that introduced the AMR task and its datasets [1, 2] have employed methods that adapt successful architectures from Video Moment Retrieval (VMR) in computer vision and train them on audio data. However, these methods still face challenges in retrieval accuracy, particularly for short moments under 10 seconds [2]. We hypothesize that this performance degradation arises because existing models can identify the approximate locations of audio moments but struggle to accurately capture their temporal boundaries.

This report proposes a system that addresses these challenges and improves robustness to diverse real-world inputs by integrating several enhancements, including a stronger feature extractor, an improved loss function, random query sampling during pretraining, and inference-time boundary refinement.

2. METHOD

Our method adopts the Unified Video Comprehension framework (UVCOM) [3] as the base architecture. We chose UVCOM because benchmark results on the CASTELLA dataset [2] show that it outperforms QD-DETR [4]. QD-DETR is used as the official DCASE Task 6 baseline, and its AMR performance is reported on the challenge website [5]. Originally proposed for Video Moment Retrieval (VMR), UVCOM is based on the Detection Transformer (DETR) architecture [6] and has strong modeling capabilities for cross-modal dependencies between audio and queries, as well as long-range temporal dependencies within audio. In this section, we describe in detail the four improvements that we introduced based on UVCOM.

2.1. Feature Extraction

We adopted M2D-CLAP [7], which integrates self-supervised learning and audio–language contrastive learning, as an advanced feature extractor for both audio and text. This model has demonstrated strong performance across a wide range of audio tasks and provides richer generalized representations than MS CLAP2023 [8], which is used in the official baseline for this task.

For audio feature extraction, we used the audio encoder of M2D-CLAP, based on ViT-Base. After converting the input audio into a mel spectrogram, the patch sequence divided into five moments along the frequency axis is fed into the encoder. The outputs of these patches are then concatenated for each time frame, yielding a feature sequence with 3840 dimensions (768 dimensions $\times$ 5 patches). Following the official recommended setting, this extraction strategy preserves structural information along the frequency axis without loss. The encoder output is obtained at a temporal resolution of approximately 6.3 fps; however, to maintain temporal consistency with the subsequent UVCOM module, we downsampled it to 1 fps by applying average pooling over each one-second interval.

For text feature extraction, we used the text encoder (GTE-base [9]) of M2D-CLAP. After tokenizing the input text, we extracted the hidden states from the final layer and used them as a 768-dimensional token sequence. We retained the outputs of all tokens, rather than only the commonly used sentence-level [CLS] token, because token-level representations are required for the cross-attention computation in UVCOM.

The extracted audio features (3840 dimensions) and text features (768 dimensions) are mapped to the same hidden dimension through linear projection layers in UVCOM. After this initial cross-modal alignment, the projected features are passed to the subsequent processing stages.

2.2. Loss Function

The overall loss function L_{total} in our method follows the formulation of UVCOM and is defined as a weighted sum of the following

losses:

$$L_{\text{total}} = \lambda_{\text{span}} L_{\text{span}} + \lambda_{\text{gIoU}} L_{\text{gIoU}} + \lambda_{\text{labels}} L_{\text{labels}} + \lambda_{\text{saliency}} L_{\text{saliency}} + \lambda_{\text{sim}} L_{\text{sim}}, \quad (1)$$

where each λ denotes the weight assigned to the corresponding loss term.

Among these losses, we focus on improving L_{labels}, which predicts the confidence score of each candidate moment. Conventional methods use cross-entropy loss for this term, but this formulation optimizes confidence scores independently of boundary accuracy, making it difficult to obtain an optimal ranking during inference. To address this issue, we introduce Varifocal Loss [10] to directly learn IoU-aware confidence scores. Varifocal Loss is formulated as follows:

$$L_{\text{VFL}} = \begin{cases} -q\left(q\log(p) + (1-q)\log(1-p)\right), & q > 0 \\ -\alpha p^{\gamma}\log(1-p), & q = 0 \end{cases} \quad (2)$$

where p denotes the confidence score predicted by the model, and q denotes the target score. For positive prediction candidates that match the ground truth ($q > 0$), we directly use the IoU between the predicted moment and the ground-truth moment as the target score q. In contrast, for negative prediction candidates that do not match the ground truth ($q = 0$), a scaling factor based on the predicted score p is introduced to reduce the contribution of easy background candidates to the loss. The parameters α and γ are hyperparameters that control the balance of the loss.

By using Varifocal Loss as L_{labels}, the model is encouraged to output higher confidence scores for prediction candidates with more accurate moment boundaries. The remaining loss terms, namely L_{span} for regression, L_{gIoU} for generalized IoU optimization, L_{saliency} for saliency-score prediction, and L_{sim} for similarity learning, are computed in the same manner as in the base UVCOM model.

2.3. Random Query Sampling for Pretraining

For model training, we follow the baseline strategy and conduct pretraining on Clotho-moment [1], followed by fine-tuning on CASTELLA [2]. Clotho-moment is a pseudo dataset constructed by processing and synthesizing the audio–caption pairs in Clotho 2.1 [11] for the AMR task. In Clotho 2.1, each audio clip is annotated with five captions, whereas Clotho-moment uses only one of them as the query.

However, training with a fixed single caption may limit the robustness of the model to linguistic variations. Therefore, our system uses Clotho 2.1, the source dataset of Clotho-moment, and adopts a strategy in which one caption is randomly sampled from the five caption candidates corresponding to each audio clip at every epoch during pretraining and then fed into the model as the query. This enables the model to learn correspondences between diverse textual expressions and audio, which is expected to improve its generalization performance.

2.4. Inference-time Boundary Refinement

The trained model outputs candidate relevant moments and the importance score of each clip, referred to as the saliency score. However, the predicted start and end times may still contain errors of several seconds, which can degrade accuracy under strict IoU thresholds. Therefore, we apply boundary refinement as an inference-time post-processing step. This procedure does not update the model weights and only adjusts the boundaries of the predicted moment.

The basic idea of boundary refinement is to use the model's initial prediction as the starting point and search for nearby start and end times such that the saliency score is high inside the moment and low outside the moment. For the prediction with the highest confidence score (top-1 prediction), we enumerate candidate moments (s, e) within a range of ± 5 seconds around the initial prediction (s_0, e_0) and evaluate them using the following score:

$$\text{score}(s,e) = C(s,e) - \alpha_{\text{ref}}\left(|s - s_0| + |e - e_0|\right) - \beta_{\text{ref}}\left|(e-s) - (e_0 - s_0)\right|. \quad (3)$$

Here, $C(s, e)$ denotes the difference between the average saliency scores inside and outside the candidate moment, and it becomes larger when the inside of the moment has higher importance than the outside. The second term penalizes the amount of displacement from the initial prediction, while the third term penalizes changes in the moment duration. We adopt the candidate (s, e) that maximizes this score, while keeping the confidence score of the moment unchanged.

The search range is fixed to ± 5 seconds, and only the penalty weights α_{ref} and β_{ref} are selected by grid search over the range from 0 to 0.1 on the validation data. The combination that achieves the highest validation mAP is used for test evaluation.

3. EXPERTIMENT

3.1. Dataset

For model training, we used Clotho-Moment and the external resource Clotho 2.1 for pretraining, and CASTELLA for fine-tuning.

Clotho 2.1 is a dataset in which each audio clip, containing acoustic events with a duration of 15–30 seconds, is annotated with five captions consisting of 8–20 words. In this study, we additionally used Clotho 2.1 as auxiliary text-augmentation data according to the strategy described in Section 2.3.

Clotho-Moment is a pseudo AMR dataset consisting of audio–caption pairs, created by synthesizing environmental sounds from Walking Tour [12] with Clotho 2.1. This dataset contains 51,230 samples.

CASTELLA is a real-world AMR dataset in which audio from YouTube videos is manually annotated. Each audio recording is 1–5 minutes long and contains multiple moments. The average caption length is 7.8 words.

For both datasets, we followed the officially provided train, validation, and test splits in our experiments. However, there were several limitations in data acquisition. CASTELLA originally consists of 1,862 audio recordings, but due to factors such as YouTube videos becoming unavailable or private, the distributed annotation files (JSONL) contained only 1,620 recordings. In this study, we used 1,561 recordings for which both the annotation and the corresponding audio source were successfully obtained (train: 834, validation: 173, test: 554). For Clotho-Moment, part of the validation set could not be obtained because of missing files in the distributed archive, and only 3,121 out of 4,107 validation samples were available. The training and test sets, consisting of 27,472 and 5,472 samples, respectively, were fully obtained.

In our experiments, these unavailable audio samples were excluded from both training and evaluation. The system was con-

Table 1: Evaluation results of the proposed method on the CASTELLA test split. Systems submitted to the challenge are denoted with SIDs 1 to 4. "VF", "M2D", "RQ", and "REF" denote Varifocal Loss, M2D-CLAP features, random query sampling, and boundary refinement, respectively. The UVCOM baseline scores are taken from the CASTELLA paper [2].

SID	System Name	checkpoint	R1@0.5	R1@0.7	mAP(avg)	mAP@0.5	mAP@0.75
*	Official baseline (QD-DETR base)	*	25.61	13.59	12.06	23.60	10.72
*	UVCOM baseline [2]	*	31.70	20.30	15.90	28.40	15.20
*	UVCOM + VF	Val best	39.20	27.62	20.53	34.43	20.42
*	UVCOM + VF + M2D	Val best	52.81	36.27	30.44	48.50	30.07
1	UVCOM + VF + M2D + RQ	Val best	54.10	37.86	31.82	50.23	30.85
2	UVCOM + VF + M2D + RQ + REF	Val best	53.95	39.68	32.28	50.34	31.66
*	UVCOM + VF	Latest	38.98	27.62	21.31	34.60	20.61
*	UVCOM + VF + M2D	Latest	53.72	37.03	30.30	48.74	29.95
3	UVCOM + VF + M2D + RQ	Latest	54.02	38.32	31.81	49.97	30.81
4	UVCOM + VF + M2D + RQ + REF	Latest	54.17	40.14	32.40	50.22	31.78

structed and evaluated using only the remaining data that were successfully acquired.

3.2. Experimental Setup

We implemented the proposed method based on UVCOM provided in the open-source Lighthouse library[1].

Feature extraction: The maximum length of the audio input was set to 300 seconds. Each audio recording was converted into a sequence with a maximum length of 300, where each element is a 3840-dimensional embedding vector, by applying M2D-CLAP encoding and average pooling as described in Section 2.1. For text queries, features were also extracted using the text encoder of M2D-CLAP and input to the model together with the audio embeddings as a 768-dimensional token sequence.

Training configuration: The model was trained for 200 epochs in both pretraining and fine-tuning. We used AdamW for network optimization and set the batch size to 32. The initial learning rate was set to 10^{-4}, and a cosine annealing scheduler was used for learning-rate decay. During training in each phase, the model was evaluated at every epoch using mAP (avg) on the validation split, and the checkpoint that achieved the highest score was saved as the validation-best model. The best model obtained during pretraining was used as the initial weight for fine-tuning. For the final system submission, we adopted both the best model after fine-tuning and the model obtained after completing all 200 epochs, referred to as the latest model.

Loss function: The weights of the loss terms in the overall loss function defined in Section 2.2 were set as follows: $L_{\text{labels}} = 4$, $L_{\text{gIoU}} = 1$, $L_{\text{span}} = 10$, $L_{\text{saliency}} = 1$, and $L_{\text{sim}} = 1$. The hyperparameters of Varifocal Loss were set to $\alpha = 0.75$ and $\gamma = 2.0$.

Multi saliency labels: In the base implementation, even when multiple ground-truth moments exist for a single query, only the first moment is used as the positive label for saliency. In contrast, our system treats all annotated moments as positive examples and uses them to compute L_{saliency}.

[1] https://github.com/line/lighthouse

4. RESULTS

Table 1 presents the evaluation results of the proposed method on the CASTELLA test dataset (test split), together with the results of the ablation study. In this experiment, we comprehensively evaluated the performance of various variants by combining the presence or absence of each proposed technical module and the model-selection criterion (Val Best / Latest).

As a result, we selected the top four configurations with high retrieval accuracy on the test data as the final submitted systems (SID 1–4). As shown in Table 1, all of these top proposed systems substantially outperformed the official baseline, achieving a maximum Recall@0.7 of 40.14.

Table 1 also confirms that each proposed technical module steadily contributes to improving retrieval performance. In particular, the performance gain obtained by introducing M2D-CLAP as the audio feature extractor was the most prominent. Finally, we submitted the inference results of each variant on the released evaluation data (evaluation set) as independent systems. The configurations and system IDs of the submitted systems are shown in Table 1.

5. CONCLUSION

In this report, we proposed a system for improving the accuracy of Audio Moment Retrieval (AMR) in long-form audio. To address the challenge of boundary localization accuracy in existing methods, our system incorporates multiple improvements, including the use of M2D-CLAP features, the introduction of Varifocal Loss, random query sampling during training, and boundary refinement during inference. Through this integrated approach, our system achieved retrieval performance that substantially outperformed the official baseline across all evaluation metrics.

6. ACKNOWLEDGMENT

We would like to thank Professors Koshinaka, Toda and Ochiai for their valuable guidance. We are also grateful to the members of our laboratory for fruitful discussions. Special thanks go to Hokuto Munakata (LY Corporation), one of the coordinators of DCASE 2026 Task 6, for providing the raw audio files of the evaluation set.

7. REFERENCES

[1] H. Munakata, T. Nishimura, S. Nakada, and T. Komatsu, "Language-based audio moment retrieval," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2025, pp. 1–5.

[2] H. Munakata, T. Imamura, T. Nishimura, and T. Komatsu, "Castella: Long audio dataset with captions and temporal boundaries," in *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026, pp. 15 352–15 356.

[3] Y. Xiao, Z. Luo, Y. Liu, Y. Ma, H. Bian, Y. Ji, Y. Yang, and X. Li, "Bridging the gap: A unified video comprehension framework for moment retrieval and highlight detection," in *Proc. CVPR*, 2024, p. 18709–18719.

[4] W. Moon, S. Hyun, S. Park, D. Park, and J.-P. Heo, "Query-dependent video representation for moment retrieval and highlight detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 23 023–23 033.

[5] DCASE Community, "Task 6: Audio moment retrieval from long audio," https://dcase.community/challenge2026/task-audio-moment-retrieval-from-long-audio, 2026, accessed: 2026-06-16.

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc European Conference on Computer Vision. (ECCV)*, 2020.

[7] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, M. Yasuda, S. Tsubaki, and K. Imoto, "M2d-clap: Masked modeling duo meets clap for learning general-purpose audio-language representation," in *Interspeech*, 2024.

[8] B. Elizalde, S. Deshmukh, and H. Wang, "Natural language supervision for general-purpose audio representations," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2024, pp. 336–340.

[9] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, "Towards general text embeddings with multi-stage contrastive learning," 2023. [Online]. Available: https://arxiv.org/abs/2308.03281

[10] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "Varifocalnet: An iou-aware dense object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8514–8523.

[11] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 736–740.

[12] S. Venkataramanan, M. N. Rizve, J. Carreira, Y. M. Asano, and Y. Avrithis, "Is imagenet worth 1 video? learning strong image encoders from 1 long unlabelled video," in *Proc. ICLR*, 2024.