# ADVANCED AUDIO MOMENT RETRIEVAL VIA CG-DETR

Technical Report

*Koki Kibata, Sayaka Yamamoto, Tomoki Kawabata, Yuma Higashino, Yuki Osawa*

Data Science Dept., Yokohama City University
22-2 Seto, Kanazawa-ku, Yokohama, Kanagawa 236-0017, Japan

## ABSTRACT

This technical report describes our system designed for the DCASE 2026 Challenge Task 6 (Audio Moment Retrieval). Utilizing the Context-Gated Detection Transformer (CG-DETR) as our structural foundation, we constructed four model variations through targeted architectural refinements, dataset expansions, and the strategic integration of both CLAP and M2D-CLAP feature backends. Within the CG-DETR encoder, we incorporated dedicated register tokens to suppress attention sinks, encouraging the model to yield noise-reduced latent representations. Concurrently, the decoder network is enhanced via Feature-wise Linear Modulation (FiLM) to inject text conditioning into the query channels, enabling dynamic time-localized predictions guided by natural language inputs. Furthermore, we shared the internal representations across the span regression and classification heads, establishing a task-aligned forecasting topology that coordinates category confidence with temporal boundaries. To optimize data pipeline transitions, we addressed inherent dataset limitations. While CASTELLA offers high-quality, long human annotations, its sample size is limited. Conversely, Clotho-Moment provides abundant but automated annotations for short 60s clips. To bridge this gap, we curated a 5-minute intermediate dataset with 14k queries for mid-phase training. During model training, we employed Periodic ASAM based on the AdamW optimizer and introduced random temporal shifting of audio data to mitigate overfitting. Additionally, adding a Sketched Isotropic Gaussian Regularization (SIGReg) loss as a feature-level penalty on the intermediate representations suppressed dimensional collapse. Following a multi-stage workflow, the model pre-trained on Clotho-Moment was iteratively fine-tuned using the CASTELLA dataset. Finally, the resulting models were blended via Model Soups, securing superior generalization without increasing inference latency.

***Index Terms—*** Audio Moment Retrieval, CG-DETR, Periodic ASAM, M2D-CLAP

## 1. INTRODUCTION

DCASE2026 Challenge Task 6 focuses on Audio Moment Retrieval (AMR) for long audio recordings. Specifically, given a long audio clip and a text query, the objective is to localize and output the precise timestamps of the moment that corresponds to the query. The challenge provides two development datasets: Clotho-Moment (a synthetic dataset) [1] and CASTELLA (a manually annotated dataset) [2], both of which include audio-text features extracted using Contrastive Language-Audio Pretraining (CLAP) [3]. Additionally, the task permits the utilization of external resources, including approved datasets and pre-trained models.

We aim to improve retrieval accuracy for this task by refining existing AMR models and implementing advanced training methodologies. To achieve even higher performance, we first replace the standard CLAP with M2D-CLAP [4] as our audio-text feature extractor. Utilizing M2D-CLAP allows our system to capture more versatile audio-language representations. In addition to these improvements, we expand our pre-training datasets specifically to bridge the gap between the pre-training and fine-tuning stages.

## 2. MODEL ARCHITECTURE

We utilize a modified version of the Correlation-Guided DEtection TRansformer (CG-DETR) [5], which has demonstrated significant success in Video Moment Retrieval (VMR)—a task closely related to Audio Moment Retrieval (AMR). In this section, we describe the specific enhancements and modifications introduced to the baseline CG-DETR framework.

### 2.1. Enhancement of Temporal Explicit Features (TEF)

CG-DETR utilizes Temporal Explicit Features (TEF) to provide temporal context. Specifically, for each token of the CLAP-encoded audio features, a 2D vector representing the normalized distances from the start and end points (scaled between 0 and 1) is concatenated. However, this implementation may not fully exploit the potential of the TEF. Because the CLAP audio features consist of hundreds of dimensions while the TEF contains only two, the TEF signal may potentially be overshadowed or ignored during the subsequent Multi-Layer Perceptron (MLP) transformation. Furthermore, due to spectral bias [6], the MLP may potentially fail to adequately capture fine-grained temporal variations.

To address these limitations, we modify the integration of TEF. First, we transform the TEF into high-dimensional Fourier features, project them via an MLP to match the dimensionality of the MLP-transformed audio features, and add them in a manner similar to standard Positional Encodings (PE). Mapping inputs to Fourier features mitigates spectral bias, enabling the network to effectively learn high-frequency, fine-grained temporal changes [7]. Additionally, we expand the TEF from 2D to 3D by appending the proportion that each token occupies relative to the entire sequence. This enhancement empowers the model to explicitly utilize temporal resolution information.

### 2.2. Dynamic Fusion of Temporal Difference Audio Features

In CG-DETR, a Transformer architecture is applied to the sequence of audio features. While this successfully captures long-range dependencies between distant tokens, it may potentially fail to ad-

equately capture fine-grained transitions between adjacent tokens. Because AMR fundamentally requires pinpointing the precise start and end boundaries of an interval corresponding to a text query, information regarding these local, frame-to-frame variations is critical.

To incorporate this crucial information, we compute the temporal differences of the MLP-transformed CLAP audio features and dynamically fuse them. Specifically, for a token at a given time step $t$, we calculate its difference from the preceding token at $t-1$ and the succeeding token at $t+1$. Subsequently, we fuse the original audio features with these temporal difference features using an attention mechanism and a Gated Linear Unit (GLU). This process enables the model to explicitly leverage the transitional dynamics between adjacent tokens. Furthermore, we apply this same temporal difference fusion to the intermediate features output by the DETR encoder. Consequently, this allows the decoder to predict the target intervals while actively factoring in the localized changes between adjacent audio tokens.

### 2.3. Integration of Register Tokens

In attention mechanisms, the sum of attention weights must equal one due to the inherent properties of the softmax function. To satisfy this constraint, the model often allocates excessive attention weights to less informative tokens or the initial token of a sequence. This phenomenon, known as "attention sinks," can become a primary source of noise.

While CG-DETR mitigates noise in the cross-attention between audio and text features by employing dummy tokens, sufficient countermeasures have not been implemented in either the Transformer encoder for text summarization or the DETR encoder. Consequently, this lack of regulation risks corrupting the text summarization with noise, or causing the DETR encoder to over-aggregate information into tokens that correspond to silent intervals.

To address these limitations, we integrate register tokens [8] into both the text-summarization Transformer encoder and the DETR Transformer encoder. Register tokens are learnable tokens appended to the input sequence that function as a temporary buffer to absorb redundant attention weights. Because these tokens are discarded immediately after the encoder computations, our approach ensures that noise-free, well-regulated features are successfully passed to the subsequent modules.

### 2.4. Text Conditioning for Content Queries

In the standard CG-DETR decoder, content queries are implemented as statically learned vectors; meaning the exact same queries are utilized regardless of the input text. Consequently, the decoder initiates its localization process without any prior context regarding the target interval, inevitably leading to an inefficient search.

To resolve this limitation, we apply Feature-wise Linear Modulation (FiLM) [9] to the content queries using text summarization features. This mechanism dynamically conditions the content queries based on the specific input text. By embedding this contextual prior, the decoder begins its search already informed about the characteristics of the interval to be extracted, thereby enabling a significantly more efficient and targeted localization process.

### 2.5. Restricting the Decoder Search Space

In the decoder of CG-DETR, the cross-attention mechanism within each layer globally scans all audio tokens. Consequently, when multiple intervals corresponding to a single text query exist within an audio sequence, the model may disperse its attention weights across all these candidate regions. This weight diffusion can potentially blur the boundaries of the predicted target intervals.

To mitigate this issue, we introduce a localized search constraint into each decoder layer by incorporating a Gaussian prior inspired by Spatially Modulated Co-Attention (SMCA) [10]. Specifically, we construct a bimodal Gaussian prior centered around the start and end boundaries of the interval, which are estimated from the positional queries input to each layer. The variance of this Gaussian distribution is dynamically determined from the content queries via a linear layer. By utilizing the resulting prior to penalize tokens that are temporally distant from the predicted boundaries, our approach enforces a more localized and precise search mechanism.

### 2.6. Enhancement of the Output Heads

In the standard CG-DETR, a span prediction head and a classification head are applied to the decoder outputs to predict the intervals corresponding to the positional and content queries, and to score how well each interval aligns with the text query for ranking. However, because there is no explicit mechanism to enforce consistency between these heads, the model may potentially output a high classification score for an interval that significantly deviates from the ground truth.

To address this discrepancy, we introduce an additional head dedicated to predicting the Intersection over Union (IoU) of the intervals, and we implement intermediate feature sharing across all heads. The inclusion of the IoU prediction head enables scale-invariant learning with respect to the target intervals and allows for a more sophisticated classification loss formulation. Furthermore, sharing intermediate features transforms the modules into task-aligned heads, implicitly guiding the training process to ensure consistency across the different output predictions.

### 2.7. Other Refinements

We replace the Parametric ReLU (PReLU) and ReLU activation functions originally utilized in CG-DETR with a parameterized Swish function that introduces a learnable parameter $\beta$. The function is defined as:

$$f(x) = x \cdot \text{sigmoid}(\beta \cdot x). \tag{1}$$

This modification effectively circumvents the dying ReLU problem and eliminates non-differentiable points, thereby enhancing the overall expressive capacity of the model.

Furthermore, we apply Layer Normalization independently to the audio and text features prior to fusing them via cross-attention. This preprocessing step ensures a more stable and efficient integration of information across the different modalities.

## 3. TRAINING STRATEGIES

During the training phase, we introduce advanced and auxiliary loss functions to enhance the model's expressive capacity and maximize overall retrieval accuracy. To further mitigate model overfitting and boost generalization capabilities, we implement audio

data augmentation techniques alongside sophisticated optimization strategies. Additionally, we introduce a specialized initialization scheme for the positional queries, which is explicitly designed to accelerate training convergence and minimize the adverse effects stemming from the domain gap between the pre-training and fine-tuning datasets.

Finally, to reduce the risk of our submitted models overfitting to the Development Datasets and to ensure robust generalization to strictly unseen data, we integrate multiple fine-tuned models using the Model Soups [11] technique.

### 3.1. Advanced Loss Formulation

In the baseline CG-DETR, the prediction heads compute a standard Cross Entropy Loss for classification and an L1 Loss for span prediction. However, because the target ground-truth intervals in the AMR task occupy a relatively small proportion of the overall audio, leading to severe class imbalance, training exclusively with Cross Entropy Loss may potentially be suboptimal. Furthermore, while the widths of ground-truth intervals vary significantly in AMR, standard L1 Loss is highly sensitive to these scale fluctuations.

To address these shortcomings, and leveraging the additionally introduced IoU prediction head, we replace the classification loss with Varifocal Loss (VFL) [12] and incorporate an L1 Loss specifically for the IoU predictions. VFL evaluates classification scores in conjunction with target IoU, effectively assigning greater loss weights to samples with high IoU. This encourages the model to output high confidence scores exclusively for candidate intervals that exhibit both correct classification and highly accurate boundaries. Moreover, introducing an L1 Loss for the IoU enables robust, scale-invariant learning regardless of the ground-truth interval lengths.

Additionally, we introduce the Sketched Isotropic Gaussian Regularization (SIGReg) [13] loss as an explicit regularization term applied to the intermediate features of our model. Specifically, this regularization is computed across multiple components: the audio features after fusing temporal differences, the standard DETR encoder output features, the encoder outputs following temporal difference fusion, the summarized text features, and the dummy tokens. The application of SIGReg implicitly guides these intermediate features to follow an isotropic Gaussian distribution, maximizing their retained information. This formulation ultimately enhances the model's expressive capacity, targeting improvements in both robustness and overall retrieval accuracy.

### 3.2. Audio Data Augmentation

During the prolonged training process of hundreds of epochs in standard CG-DETR, the model is consistently fed identical audio feature sequences. Under this configuration, the model may potentially engage in shortcut learning with respect to temporal positions. Specifically, rather than properly associating a given text query with the relevant audio features, the model may potentially learn to map the queries directly to the positional encodings.

To address this vulnerability, we modify the training process to input randomly shifted audio feature sequences into the model. When applying this random shift, the ground-truth target intervals are correspondingly shifted by the exact same offset. This mechanism robustly suppresses inappropriate shortcut learning. Furthermore, while this procedure effectively increases data variance, the total number of training samples remains unchanged, resulting in virtually no additional computational overhead during training.

### 3.3. Optimization Strategies

CG-DETR employs AdamW as its optimizer and StepLR as its learning rate scheduler. However, this optimization strategy does not necessarily guarantee convergence to a highly generalizable solution.

To address this, we adopt an AdamW-based Adaptive Sharpness-Aware Minimization (ASAM) [14] as our optimizer. ASAM is generally known for its ability to find flat minima, which are strongly associated with better generalization performance. By introducing the concept of adaptive sharpness to standard Sharpness-Aware Minimization (SAM) [15], ASAM enables the search for flat minima that are scale-invariant with respect to the model parameters.

A notable drawback of ASAM, however, is that it requires twice the number of gradient computations per training step compared to standard AdamW. To circumvent this issue, rather than applying ASAM at every single update step, we implement a periodic configuration where ASAM is applied only once every several steps. This strategy allows us to effectively search for highly generalizable solutions while keeping the computational overhead manageable.

Furthermore, we replace the learning rate scheduler with a cosine learning rate decay [16]. This adjustment facilitates more effective optimization over a prolonged number of training epochs. Additionally, we introduce a warmup phase that gradually increases the learning rate at the beginning of training. This mitigates gradient instability during the initial stages, ensuring a more stable and robust learning process overall.

### 3.4. Initialization of Positional Queries

In standard CG-DETR, the positional queries are initially configured as random parameters. Furthermore, at the onset of fine-tuning, the parameters learned during pre-training are transferred directly, alongside the rest of the network weights. These default configurations may potentially hinder training convergence during both the pre-training and fine-tuning stages. More critically, if a domain gap exists between the pre-training and fine-tuning datasets, this direct transfer may potentially result in negative transfer.

To address these issues, we initialize the positional queries based on insights derived from a statistical analysis of the training dataset. First, we prepare $3 \times N$ positional queries and distribute them evenly across $N$ distinct locations, assigning 3 queries to each location. Subsequently, for the positional queries at each location, we assign values approximating the 15th, 50th, and 85th percentiles of the widths of all ground-truth intervals present in the training dataset. Moreover, during the fine-tuning phase, rather than transferring the positional query parameters from the pre-trained model, we independently re-apply this statistical initialization strategy based on the fine-tuning dataset. This approach effectively prevents negative transfer while improving overall training convergence.

### 3.5. Integration via Model Soups

Among the candidate models generated through multiple fine-tuning runs, the one achieving the highest performance metrics on the validation or test splits of the Development Datasets does not necessarily possess the best generalization capabilities. In other words, a model that performs best on the Development Datasets may potentially fail to perform optimally on strictly unseen data. Consequently, relying exclusively on the Development Datasets to

select the final model for submission carries a potential risk of overfitting to the local data distribution.

To mitigate this risk, we integrate all models obtained across multiple fine-tuning iterations using Model Soups [11] to achieve a robust ensemble effect. Model Soups is a technique that averages the weights of multiple fine-tuned models into a single model. Unlike traditional ensembling methods that require multiple forward passes, this weight-averaging approach yields the performance benefits of an ensemble without incurring any additional computational overhead during inference. While it is common practice to integrate models using a greedy strategy (Greedy Soups), we primarily adopt Uniform Soups to deliberately reduce our reliance on the Development Datasets. Furthermore, we also employ Fisher-Weighted Averaging (FWA) Soups, which leverages Fisher information computed from non-training data, to achieve a more informed and statistically balanced parameter integration.

## 4. ADVANCED FEATURE EXTRACTION

We explore to replace the canonical CLAP [3] feature extractor with M2D-CLAP [4], which integrates CLAP with Masked Modeling Duo (M2D) [17]. Because M2D-CLAP extracts more versatile audio-language representations, its application may potentially contribute to substantial improvements in overall retrieval accuracy for the AMR task.

## 5. PRE-TRAINING DATASET EXPANSION

To explicitly bridge the domain gap between the pre-training and fine-tuning datasets, we constructed a novel synthetic pre-training dataset. The specific distribution details are provided in Table 1.

Table 1: Synthetic Pre-training Dataset Specifications

| **Metric / Parameter** | **Value / Configuration** |
|---|---|
| Total Training Audios | 5,000 clips |
| Total Training Text Queries | 14,385 queries |
| Total Validation Audios | 500 clips |
| Total Validation Text Queries | 1,414 queries |
| Absolute Clip Duration | 5 minutes (Fixed) |
| Acoustic Events per Clip | 1 to 5 events |

The dataset was generated by mixing clean acoustic events over diverse background soundscapes. Background tracks were sourced from the DEMAND database, filtering out files with excessive event noise through manual listening tests to retain 6 clean environmental categories. Acoustic event samples were extracted from WavCaps [18], utilizing subsets from AudioSet, BBC Sound Effects, and Sound Bible. Although certain AudioSet IDs overlapped with the Clotho-Moment blacklist, they were retained since our background mixing and random placement strategies alter the audio contexts, mitigating data leakage concerns.

## 6. SUBMITTED MODEL VARIATIONS

We submit the results of four distinct model variations that integrate all the aforementioned architectural enhancements and training strategies.

The first variation presents the results of a model trained and inferred using the standard CLAP features provided in the official Development Datasets. The second variation reflects the results obtained by replacing the baseline CLAP features with M2D-CLAP features for both the training and inference stages. The third variation utilizes the M2D-CLAP features and incorporates a two-stage pre-training pipeline using our additionally constructed pre-training dataset prior to the fine-tuning phase. For these first three variations, the fine-tuning process was executed 25 times using different random seeds. Crucially, during this phase, both the train and test splits of the CASTELLA dataset were combined into a single, unified dataset and treated entirely as training data. The models obtained from these 25 individual runs were subsequently integrated using the Uniform Soups method.

Finally, the fourth variation adopts the exact same training configuration as the third variation, but employs FWA Soups instead of Uniform Soups during the final model integration phase.

Additionally, across all submitted configurations, the DETR decoder is uniformly configured to utilize 30 queries.

Table 2: Evaluation results on the CASTELLA test split (reference). The notation R@1(0.7) denotes the Recall@1 metric evaluated at an IoU threshold of 0.7. Furthermore, the baseline performance presented here reflects the official benchmark values published on the competition website.

| Model | R@1(0.7) | R@1(0.5) | R@5(0.7) | R@5(0.5) |
|---|---|---|---|---|
| Baseline | 13.59 | 25.61 | N/A | N/A |
| Ours 1 | 26.80 | 39.50 | 41.43 | 57.83 |
| Ours 2 | 40.70 | 55.98 | 59.98 | 76.85 |
| Ours 3 | 42.51 | 58.32 | 60.74 | 78.06 |
| Ours 4 | 42.28 | 58.17 | 60.14 | 78.06 |

For reference, this report also provides the evaluation results on the CASTELLA test split (Table 2). These model were trained exclusively on the CASTELLA train split, where we executed 10 independent fine-tuning runs and subsequently integrated the resulting models prior to evaluation. The results demonstrate that the architectural enhancements to CG-DETR, the advanced training strategies, the utilization of M2D-CLAP, and the expansion of the pre-training dataset all effectively contribute to enhanced retrieval accuracy. Conversely, the evaluation also reveals that the performance difference between FWA Soups and Uniform Soups is marginal. Furthermore, an analysis comparing Recall@1 and Recall@5, along with the results across varying IoU thresholds, suggests that further accuracy gains could be achieved by refining the predicted intervals and implementing more sophisticated re-ranking mechanisms.

## 7. CONCLUSION

In this report, we detailed our comprehensive approach for DCASE2026 Challenge Task 6, which encompasses architectural enhancements to the CG-DETR framework, the implementation of advanced training strategies, the integration of a superior feature extractor, and the expansion of the pre-training dataset. Furthermore, we demonstrated the efficacy of these proposed modifications, establishing that they collectively yield a substantial improvement in retrieval accuracy over the official baseline. Finally, our analysis suggests that further performance gains could be achieved through future refinements, specifically by employing post-processing techniques such as the re-ranking of predicted intervals.

## 8. 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," 2026.

[3] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[4] 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*, 2024, pp. 57–61.

[5] W. Moon, S. Hyun, S. Lee, and J.-P. Heo, "Correlation-guided query-dependency calibration in video representation learning for temporal grounding," *arXiv preprint arXiv:2311.08835*, 2023.

[6] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 5301–5310.

[7] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in neural information processing systems*, vol. 33, pp. 7537–7547, 2020.

[8] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," in *International conference on learning representations*, vol. 2024, 2024, pp. 2632–2652.

[9] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[10] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of detr with spatially modulated co-attention," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3621–3630.

[11] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International conference on machine learning*. PMLR, 2022, pp. 23 965–23 998.

[12] 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*, 2021, pp. 8514–8523.

[13] R. Balestriero and Y. LeCun, "Lejepa: Provable and scalable self-supervised learning without the heuristics," *arXiv preprint arXiv:2511.08544*, 2025.

[14] J. Kwon, J. Kim, H. Park, and I. K. Choi, "Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks," in *International conference on machine learning*. PMLR, 2021, pp. 5905–5914.

[15] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *International Conference on Learning Representations*.

[16] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017.

[17] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked modeling duo: Towards a universal audio pre-training framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2391–2406, 2024.

[18] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 3339–3354, 2024.