COARSE-TO-FINE AUDIO MOMENT RETRIEVAL WITH TEMPORAL REFINEMENT AND RE-RANKING

Technical Report

Óscar Calvet, Doroteo T. Toledano

Audio, Data Intelligence and Speech research group (AUDIAS)
Escuela Politecnica Superior
Universidad Autonoma de Madrid
oscar.calvet@estudiante.uam.es

ABSTRACT

This technical report describes our submission to Task 6 of the DCASE 2026 Challenge, which addresses audio moment retrieval from long audio recordings. The goal of the task is to localize the temporal segment, or segments, that match a natural-language query in an untrimmed audio recording. Our main approach follows a coarse-to-fine retrieval strategy based on UVCOM-style temporal localization models and window-level audio embeddings. First, a coarse model processes the full audio recording and produces a ranked set of candidate moments. Then, a second refinement model operates on local crops around the most promising candidates using higher-resolution audio features, improving the temporal precision of the predicted boundaries. Finally, a lightweight candidate reranker combines temporal, confidence, audio-text similarity, and boundary-context features to select and rank the final predictions. We also apply submission-oriented postprocessing, including timestamp rounding, duration clamping, duplicate removal, and boundary clipping. Our best single system achieves an R1@0.7 of 40.01, while our ensemble system achieves an R1@0.7 of 42.09. These results show that combining global proposal generation, local high-resolution refinement, and candidate-level reranking is an effective strategy for language-based audio moment retrieval in long recordings.

***Index Terms*—** Audio moment retrieval, language-based audio retrieval, text-to-audio retrieval, long audio understanding, temporal localization, cross-modal retrieval, coarse-to-fine localization, DETR-based retrieval

1. INTRODUCTION

Language-based audio retrieval aims to search audio content using free-form natural-language queries. While clip-level audio-text retrieval determines whether a recording is relevant, many applications require finer temporal localization: the system must identify when the queried event occurs within a long, untrimmed signal. This is useful for locating highlights in broadcasts, finding events in surveillance recordings, or navigating multimedia archives.

Audio moment retrieval (AMR) addresses this setting by taking a long recording and a natural-language query as input and returning one or more temporal windows matching the described sound event or scene. Unlike sound event detection, AMR is not limited to a predefined taxonomy; it requires open-vocabulary matching between complex textual descriptions and localized acoustic content. The system must therefore jointly model audio-text semantic alignment and temporal structure over long sequences.

DCASE 2026 Challenge Task 6 focuses on AMR from long audio recordings and is closely related to Clotho-Moment [1] and CASTELLA [2]. Clotho-Moment introduced a simulated AMR framework and showed the advantage of DETR-style [3] temporal localization over sliding-window clip retrieval, while CASTELLA extends the task to real-world one-to-five-minute recordings with human-annotated captions and boundaries. These benchmarks highlight key difficulties: target moments can be short, repeated, overlapping, or embedded in complex backgrounds. Pretrained text-to-audio encoders provide useful semantic representations, but they are typically trained for complete clip-caption matching rather than accurate boundary prediction, so they must be adapted to a temporal localization framework.

Our submission uses a coarse-to-fine strategy. A global UVCOM-style [4] model first identifies promising temporal regions in the full recording. A second UVCOM-style refinement model then operates on local high-resolution crops around the top candidates to improve boundary precision. Finally, a lightweight candidate reranker scores the refined candidates using temporal, confidence, audio-text similarity, and boundary-context features. We submit three single systems and one ensemble system based on this pipeline.

2. SYSTEM OVERVIEW

The submitted systems formulate audio moment retrieval as open-vocabulary temporal localization in long audio. Given an audio recording represented by a sequence of window-level audio embeddings and a natural-language query represented by a text embedding, the system predicts a ranked set of temporal windows. Each prediction is a segment

$$p_i = [s_i, e_i], \tag{1}$$

with an associated confidence score c_i. The reference annotation is denoted as

$$g = [s_g, e_g], \tag{2}$$

where s_g and e_g are the start and end times of the target moment.

The main pipeline is trained and applied in stages. A coarse UVCOM model first performs global localization over the full audio

recording. A Fine-UVCOM model then refines the most promising coarse candidates using local high-resolution crops. A candidate reranker finally scores the refined candidates using candidate-level temporal, semantic, confidence, and boundary-context features. The final predictions are postprocessed before submission. The complete system is therefore not trained end-to-end; instead, each stage is optimized after the previous stage has generated the inputs required for the next one.

2.1. Temporal overlap

Temporal overlap between a prediction $p = [s_p, e_p]$ and a reference segment $g = [s_g, e_g]$ is measured with temporal intersection over union. When span regression is performed in center-width coordinates, each segment is represented by its midpoint and its duration, computed from the corresponding start and end times. The localization losses can be applied either in start-end coordinates or in normalized center-width coordinates, depending on the corresponding UVCOM implementation.

2.2. Coarse UVCOM localization

The first stage performs global retrieval over the full audio recording. Given a coarse audio feature sequence A^{coarse} and a query representation z_q, the coarse UVCOM [4] model predicts a fixed-size set of candidate moments:

$$P = \{(p_i, c_i)\}_{i=1}^{N}. \tag{3}$$

The model follows a DETR-style set prediction formulation. Instead of scoring all possible start-end pairs, it uses a fixed set of proposal slots and predicts temporal spans and confidence scores.

Training uses bipartite matching between predicted proposals and ground-truth moments. Let G denote the set of reference moments for the query. The optimal assignment is obtained by minimizing a matching cost:

$$\sigma^* = \arg\min_{\sigma} \sum_i \mathcal{C}_{\text{match}}\left(p_i, g_{\sigma(i)}\right). \tag{4}$$

The matching cost combines classification, span regression, and generalized temporal IoU terms:

$$\mathcal{C}_{\text{match}}(p_i, g) = \lambda_{\text{cls}}\mathcal{C}_{\text{cls}}(c_i) + \lambda_{\ell_1}\|p_i - g\|_1 + \lambda_{\text{giou}}\mathcal{C}_{\text{giou}}(p_i, g), \tag{5}$$

where the generalized temporal IoU cost is

$$\mathcal{C}_{\text{giou}}(p_i, g) = 1 - \text{GIoU}(p_i, g). \tag{6}$$

After matching, the coarse model is optimized with the same three components:

$$\mathcal{L}_{\text{coarse}} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\ell_1}\mathcal{L}_{\ell_1} + \lambda_{\text{giou}}\mathcal{L}_{\text{giou}}. \tag{7}$$

The classification term is a cross-entropy loss over foreground and background proposal labels:

$$\mathcal{L}_{\text{cls}} = \text{CE}(c_i, y_i), \tag{8}$$

where matched predictions are assigned foreground labels and unmatched predictions are assigned background labels. The span regression term penalizes start and end boundary errors using

$$\mathcal{L}_{\ell_1} = \|p_i - g\|_1. \tag{9}$$

The generalized temporal IoU loss penalizes poor temporal overlap and poor interval placement:

$$\mathcal{L}_{\text{giou}} = 1 - \text{GIoU}(p_i, g). \tag{10}$$

For temporal intervals, generalized IoU is defined as

$$\text{GIoU}(p, g) = \frac{|I|}{|U|} - \frac{|C| - |U|}{|C|}, \tag{11}$$

where C is the smallest enclosing interval containing both p and g, and U is their union. The enclosing interval length is

$$|C| = \max(e_p, e_g) - \min(s_p, s_g), \tag{12}$$

and the union length is

$$|U| = (e_p - s_p) + (e_g - s_g) - |I|, \tag{13}$$

where

$$|I| = \max\left(0, \min(e_p, e_g) - \max(s_p, s_g)\right). \tag{14}$$

This term penalizes low overlap and also accounts for the distance between non-overlapping or weakly overlapping predictions.

At inference, the coarse model produces a ranked set of candidate windows. The top K candidates are retained and passed to the local refinement stage.

2.3. Fine-UVCOM local refinement

The second stage improves the temporal precision of the coarse candidates. While the coarse model operates over the full recording, the Fine-UVCOM model operates over local crops centered around the coarse predictions. This allows the refinement model to use higher-resolution audio features and to focus its capacity on boundary correction.

For each coarse candidate $r = [s_r, e_r]$, a local crop is defined by adding temporal context:

$$t_{\text{start}} = \max(0, s_r - \delta), \qquad t_{\text{end}} = \min(D, e_r + \delta), \tag{15}$$

where D is the audio duration and δ is the context size. In the submitted systems, $\delta = 5$ seconds. The ground-truth span is transformed into the crop-local time axis by expressing its start and end times relative to the crop start. Fine-UVCOM then predicts a span in this local time axis, and the predicted boundaries are mapped back to global recording time by adding the crop start time.

The Fine-UVCOM refiner uses the same UVCOM detection loss as the coarse model, but applies it in local crop coordinates. For crops selected as span positives, the localization objective is

$$\mathcal{L}^{+}_{\text{UVCOM}} = \lambda_{\text{cls}}\mathcal{L}^{\text{local}}_{\text{cls}} + \lambda_{\ell_1}\|p^{\text{local}} - g^{\text{local}}\|_1 + \lambda_{\text{giou}}\mathcal{L}^{\text{local}}_{\text{giou}}. \tag{16}$$

For top-K refinement, each coarse proposal defines a local crop. We denote the temporal span of the k-th proposal, and therefore the k-th crop candidate, as c_k. Crops are annotated according to their overlap with the ground-truth moment. The refiner also predicts a candidate-level confidence score s_k, supervised by the proposal IoU:

$$\mathcal{L}_{\text{cand}} = \text{BCE}\left(s_k, \text{IoU}(c_k, g)\right). \tag{17}$$

Crops with no overlap, or proposal IoU below a small threshold, receive background supervision:

$$\mathcal{L}_{\text{neg}-\text{bg}} = \text{CE}(c_i, \text{background}). \tag{18}$$

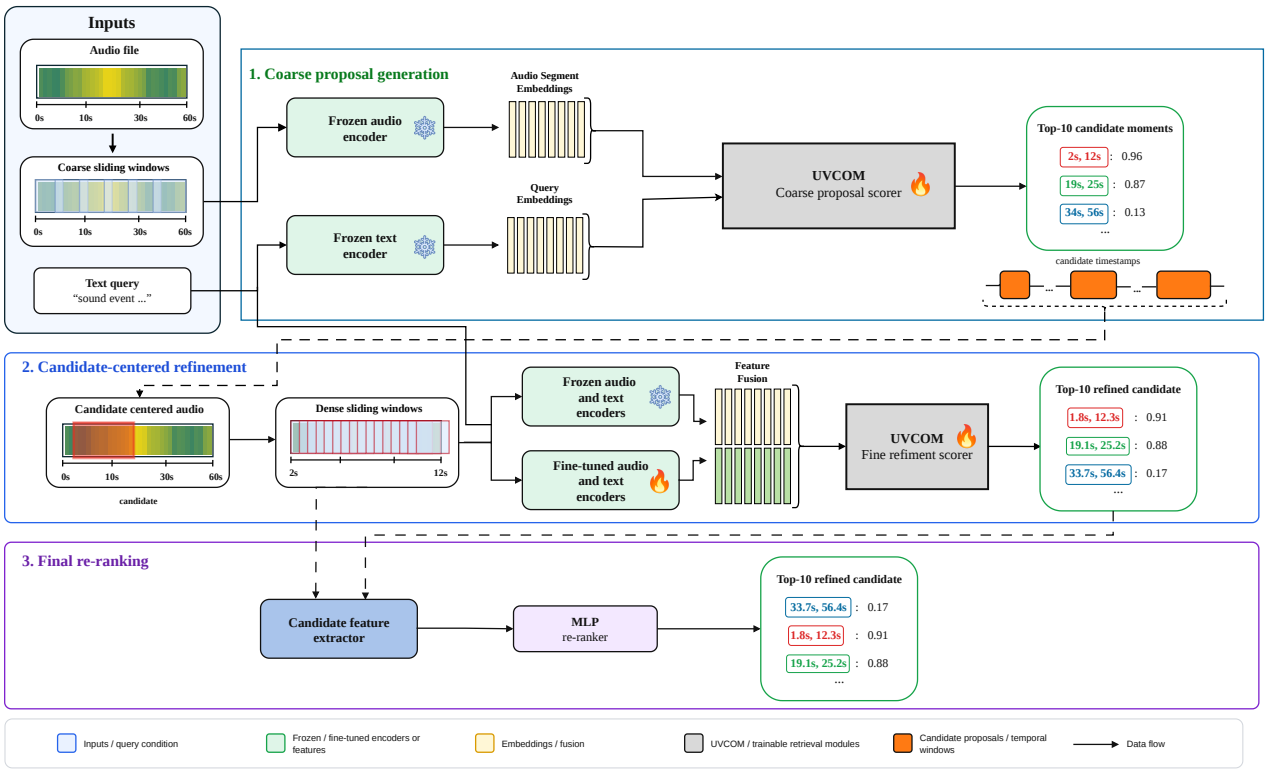

Figure 1: Overview of the proposed audio moment retrieval pipeline.

Finally, candidates belonging to the same original query are trained with a top-K selection loss:

$$\mathcal{L}_{\text{topk}} = \text{CE}\left([s_1, \ldots, s_K], \arg\max_k \text{IoU}(c_k, g)\right). \tag{19}$$

The complete fine-stage objective is

$$\mathcal{L}_{\text{fine}} = \mathcal{L}^{+}_{\text{UVCOM}} + \lambda_{\text{cand}}\mathcal{L}_{\text{cand}} + \lambda_{\text{neg}}\mathcal{L}_{\text{neg-bg}} + \lambda_{\text{topk}}\mathcal{L}_{\text{topk}}. \tag{20}$$

The most effective refinement setup is trained on target-domain local crops so that the refinement distribution matches the evaluation domain.

2.4. Candidate reranking

After local refinement, each query is associated with a list of top-K refined candidates:

$$\mathcal{C} = \{p_1, \ldots, p_K\}. \tag{21}$$

The detector confidence is useful but does not always rank candidates according to their temporal overlap with the reference. A separate reranker is therefore trained to select the best candidate from the refined list.

For each candidate, the target quality is defined by its temporal IoU with the ground truth, $u_i = \text{IoU}(p_i, g)$, and the oracle-best candidate is $i^* = \arg\max_i u_i$. The reranker maps a candidate feature vector x_i to a scalar score $r_i = f_\theta(x_i)$. The feature vector contains candidate temporal descriptors, localization confidence scores, pooled audio features within and around the candidate, audio-text similarity evidence, and boundary-context information.

The reranker is trained with soft labels derived from candidate IoUs:

$$y_i = \frac{\exp(u_i/\tau)}{\sum_{j=1}^{K}\exp(u_j/\tau)}, \tag{22}$$

where τ is the soft-label temperature. The corresponding loss is

$$\mathcal{L}_{\text{soft}} = -\sum_{i=1}^{K} y_i \log \frac{\exp(r_i)}{\sum_{j=1}^{K}\exp(r_j)}. \tag{23}$$

The reranker also includes a pairwise ranking term. For pairs whose IoU values differ by at least m_{iou}, the score of the better candidate is encouraged to exceed the score of the worse candidate:

$$\mathcal{L}_{\text{pair}} = \sum_{i=1}^{K}\sum_{j=1}^{K} \mathbf{1}\left[u_i > u_j + m_{\text{iou}}\right] \max\left(0, -(r_i - r_j)\right). \tag{24}$$

This formulation prevents candidates with similar IoU values from being treated as equally incorrect. The final reranker loss is

$$\mathcal{L}_{\text{reranker}} = \mathcal{L}_{\text{soft}} + \lambda_{\text{pair}}\mathcal{L}_{\text{pair}}, \tag{25}$$

2.5. Inference-time score blending

During inference, the reranker score can be combined with the detector score to improve calibration. Let d_i denote the Fine-UVCOM score for candidate p_i. The final score is computed additively:

$$\text{score}_i^{\text{final}} = r_i + \alpha d_i. \tag{26}$$

This operation is used only at inference time and is not part of the training objective.

Table 1: Results for the submitted EAT, BEATs, and EAT+BEATs systems at each stage of the pipeline evaluated on CASTELLA test.

Stage	Model	Recall1@0.5	Recall1@0.7	mAP (avg)	mAP@0.5	mAP@0.75
Coarse	EAT	42.95	28.64	22.68	39.64	21.34
	BEATs	47.22	31.33	23.92	41.56	22.56
	EAT+BEATs	46.40	31.55	23.91	42.19	22.20
Fine	EAT	47.29	34.45	25.33	41.61	24.86
	BEATs	46.18	33.26	24.06	40.34	23.81
	EAT+BEATs	47.51	35.04	25.39	41.93	24.74
Reranker	EAT	56.12	40.01	30.17	47.16	30.39
	BEATs	53.30	36.82	28.00	45.96	27.41
	EAT+BEATs	54.27	37.56	29.66	47.42	29.53
Ensemble	Ensemble	**59.39**	**42.09**	**33.64**	**48.73**	**34.03**

2.6. Stage-wise optimization

The complete pipeline is optimized sequentially. First, the coarse UVCOM model is trained on full-recording features with $\mathcal{L}_{\text{coarse}}$. The trained coarse model is then frozen and used to generate top-K candidate windows. These windows define local crops for training Fine-UVCOM with $\mathcal{L}_{\text{fine,total}}$. After refinement training, both localization models are frozen and used to generate refined candidate lists. The reranker is finally trained on these lists using $\mathcal{L}_{\text{reranker}}$. Finally we adjust the value α to maximize R1@0.7.

At inference time, the same ordering is followed: coarse global localization, local refinement, candidate reranking, score blending, and final postprocessing. The ensemble submission combines the outputs of the submitted single systems at inference time to improve robustness, without changing the training objectives of the individual systems.

3. EXPERIMENTAL SETUP

Audio-text pretraining used Clotho [5], TACOS weak [6], AudioCaps [7], and WavCaps [8] to train EAT–RoBERTa [9, 10] and BEATs–RoBERTa [11, 10] models. We followed the curriculum in [12] and added the cross-encoder estimated-correspondence stage from [13], using an auxiliary PaSST–RoBERTa model [14, 10]. Audio was split into non-overlapping 10-second segments, independently embedded, mean-pooled, and trained with a bidirectional contrastive objective for 20 epochs, with batch sizes 24 (EAT) and 20 (BEATs), temperature 0.05, AdamW weight decay 10^{-4}, one warmup epoch, and a cosine schedule from 2×10^{-5} to 10^{-7}. These CASTELLA-unadapted embeddings were used for all coarse-stage models. The encoders were then fine-tuned for the fine stage with temporal supervision from TACOS strong and CASTELLA. Audio was represented with 2-second windows and a 0.5-second hop; embeddings overlapping each annotated moment were pooled and aligned with the query using the TACOS moment-level contrastive loss [6]. Fine-UVCOM receives both representations by concatenating the CASTELLA-unadapted embeddings with the temporally fine-tuned embeddings.

For coarse localization, UVCOM was trained on full recordings from TACOS strong, Clotho-Moments, and CASTELLA using 2-second windows with a 1-second hop. We trained EAT, BEATs, and EAT+BEATs models with 10 proposal slots and retained the top $K = 10$ candidates for refinement. All coarse models used the classification, L1, and temporal GIoU losses from Section 2, with training and Hungarian matching weights $\lambda_{\text{cls}} = 4$, $\lambda_{\ell_1} = 10$, and $\lambda_{\text{giou}} = 1$. We used AdamW with learning rate 10^{-4}, weight decay 10^{-4}, batch size 128 for EAT and BEATs, batch size 96 for EAT+BEATs, and trained for 300 epochs.

Fine-UVCOM was trained only on CASTELLA crops generated from coarse candidates. Each crop added 5 seconds of context on both sides of the coarse prediction and was clipped to the audio duration. The refiner used 2-second windows with a 0.5-second hop and concatenated non-fine-tuned and fine-tuned encoder features; for EAT+BEATs this concatenates pretrained and fine-tuned EAT and BEATs features. Training used crop-local coordinates, the same localization weights as the coarse model, and span-positive supervision from the best-IoU crop among the top-K crops for each query. Crops with no ground-truth overlap or proposal IoU $\leq \theta_{\text{neg}} = 0.05$ were supervised as background. The auxiliary loss weights were $\lambda_{\text{cand}} = \lambda_{\text{neg}} = \lambda_{\text{topk}} = 1$, and the remaining training configuration matched coarse localization.

The candidate reranker was trained using the top $K = 10$ refined candidates for each query. Each candidate was represented with temporal features, detector confidence features, audio-text similarity features, and context/boundary features. The temporal features included start time, end time, duration, log-duration, normalized position, center time, and width. The detector features included coarse and Fine-UVCOM scores. The audio-text features included pooled audio embeddings inside the candidate, the query embedding, and cosine similarities between them. Context and boundary features were computed by pooling audio embeddings in the left context, right context, start-boundary region, and end-boundary region.

The reranker was implemented as an MLP with LayerNorm on the input, GELU activations, dropout, and a final linear layer producing a scalar candidate score. All rerankers used dropout rate 0.1, batch size 1 query group with up to 10 candidates per group, AdamW optimization, learning rate 1×10^{-3}, and weight decay 1×10^{-4}. The EAT and EAT+BEATs rerankers used MLP shape [30, 128, 128, 1], while the BEATs reranker used MLP shape [30, 64, 64, 1]. The EAT and BEATs rerankers were trained for 120 epochs, and the EAT+BEATs reranker was trained for 80 epochs. The training objective used the soft IoU-label ranking loss combined with the pairwise ranking loss. The soft-label temperature was $\tau = 0.10$ for EAT and $\tau = 0.05$ for BEATs and EAT+BEATs. For EAT and EAT+BEATs, the pairwise loss weight was $\lambda_{\text{pair}} = 1.0$ and the IoU margin was $m_{\text{iou}} = 0.05$; for BEATs, $\lambda_{\text{pair}} = 1.5$ and $m_{\text{iou}} = 0.07$.

At inference time, each single system generated 10 coarse candidates, refined them with Fine-UVCOM, reranked the refined candidates, and applied final postprocessing. When used, score blending added the Fine-UVCOM detector score to the reranker score with $\alpha = 0.40$ for EAT, $\alpha = 0.30$ for BEATs, and $\alpha = 0.50$ for EAT+BEATs.

An ensemble model was created by merging the candidate lists produced by the three single systems. Specifically, we took the top 10 candidates from EAT, BEATs, and EAT+BEATs, yielding at most $K_{\text{ens}} = 30$ candidates per query. Within each source system, raw candidate scores were normalized with a z-score transformation and passed through a sigmoid before merging. Exact duplicate windows were removed using identical rounded start–end keys. A separate learned MLP reranker was then trained to score the merged top-30 candidate set, using MLP shape [57, 128, 128, 1] with LayerNorm on the 57-dimensional input, GELU activations, and dropout rate 0.1 after each hidden layer. The ensemble reranker used AdamW optimization with learning rate 1×10^{-3} and weight decay 1×10^{-4}, $\tau = 0.05$, $\lambda_{\text{pair}} = 1.5$, and $m_{\text{iou}} = 0.05$. The ensemble final score was an additive blend of the ensemble reranker score and proposal score with $\alpha = 0.10$.

The final predictions were rounded to the nearest second, constrained to a minimum duration of one second, deduplicated, and clipped to the audio duration.

4. RESULTS

Table 1 reports the performance of the submitted systems on the CASTELLA test set. The coarse UVCOM stage provides the initial localization results, with BEATs and EAT+BEATs obtaining the strongest coarse R1@0.7 values. Fine-UVCOM improves temporal localization for all systems, reaching an R1@0.7 of 35.04 with EAT+BEATs. The reranker further improves all systems, with the EAT reranker achieving the best single-system result, R1@0.7 of 40.01 and mAP@0.75 of 30.39. The ensemble provides the best overall performance, reaching R1@0.7 of 42.09, mAP of 33.64, and mAP@0.75 of 34.03.

5. CONCLUSIONS

This technical report described a stage-wise audio moment retrieval pipeline that combines global proposal generation, local boundary refinement, and candidate-level ranking. The approach separates long-recording search from high-resolution temporal adjustment, allowing each component to focus on a different part of the localization problem. The submitted systems demonstrate that this coarse-to-fine design is effective for CASTELLA, with the ensemble of EAT, BEATs, and EAT+BEATs providing the strongest final submission.

6. ACKNOWLEDGMENT

This research was supported by project PID2024-160789OB-I00 funded by MICIU/AEI/10.13039/501100011033 / FEDER, UE. Computational resources for this research were partly provided by CCC-UAM.

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," 2026.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *CoRR*, vol. abs/2005.12872, 2020. [Online]. Available: https://arxiv.org/abs/2005.12872

[4] 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," 2023. [Online]. Available: https://arxiv.org/abs/2311.16464

[5] K. Drossos, S. Lipping, and T. Virtanen, "Clotho dataset," May 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4783391

[6] P. Primus, F. Schmid, and G. Widmer, "Tacos: Temporally-aligned audio captions for language-audio pretraining," 2025. [Online]. Available: https://arxiv.org/abs/2505.07609

[7] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *NAACL-HLT*, 2019.

[8] 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, p. 3339–3354, 2024. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2024.3419446

[9] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "EAT: Self-supervised pre-training with efficient audio transformer," in *IJCAI*, 2024.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[11] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," 2022. [Online]. Available: https://arxiv.org/abs/2212.09058

[12] O. Calvet and D. Torre Toledano, "Cross-modal attention architectures for language-based audio retrieval," in *Proceedings of the 10th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2025)*, Barcelona, Spain, October 2025, pp. 110–114.

[13] P. Primus, F. Schmid, and G. Widmer, "Estimated audio-caption correspondences improve language-based audio retrieval," 2024. [Online]. Available: https://arxiv.org/abs/2408.11641

[14] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*. ISCA, 2022, pp. 2753–2757. [Online]. Available: https://doi.org/10.21437/Interspeech.2022-227