TEXT-SPACE IMAGINATION OF AUDIO RETRIEVAL VIA JOINT-SPACE PROJECTION

Technical Report

Chao-Han Huck Yang[1*] *Zih-Ching Chen*[1] *Eli Chien*[2] *Sabato Marco Siniscalchi*[3,4]

[1]NVIDIA [2]National Taiwan University [3]University of Palermo [4]Georgia Tech
{hucky,virginiac}@nvidia.com

ABSTRACT

We study audio moment retrieval (AMR) as *text-space imagination of audio*: given a natural-language query and a long recording, a contrastive audio–language model can locate when the described sound is active by scoring each second of audio against the text in a shared embedding space. DCASE 2026 Task 6 provides pre-extracted MS-CLAP-2023 *backbone* features (768 dimensions) rather than raw audio. We first show, through a matched versus mismatched query analysis, that cosine similarity in this backbone space carries almost no query-specific signal: a query's peak relevance on its own audio is no sharper than that of a random query. Our method, **JointProj**, restores the missing cross-modal alignment by passing the provided features and a re-encoded query through MS-CLAP's own projection heads into the 1024-dimensional joint space, where cosine similarity yields a sharp per-second relevance curve. Moments are then localized by full-width-at-half-maximum (FWHM) peak detection. On the Clotho-Moment validation split, scored with the official tool, JointProj raises Recall1@0.7 from 13.2 to 46.0 and mAP from 12.4 to 42.5 over backbone cosine. We submit four systems: three FWHM peak-width variants that perform best in different moment-length regimes, and one exploratory localizer in which a language model reasons over the joint-space curve. We select the strongest development configuration for the blind evaluation.

***Index Terms*—** audio moment retrieval, audio language modeling, joint-space projection, text-space imagination

1. INTRODUCTION

Audio moment retrieval asks, for a free-form text query and a long recording, for the time interval(s) where the described event occurs [1]. In this work, we aim to frame it as *text-space imagination of audio*: a contrastive language–audio model embeds text and audio in a shared space, so a query can be projected onto time by scoring every one-second window of audio against the query embedding. The peaks of this per-second relevance curve are the model's imagination of when the event happens.

DCASE 2026 Task 6 [2] provides pre-extracted MS-CLAP-2023 features [3] instead of raw audio: per-second *audio* embeddings and per-token *text* embeddings, both 768 dimensions. These are *backbone* (pre-projection) features, whereas MS-CLAP's contrastive alignment lives in a separate 1024-dimensional joint space reached through learned projection heads. Cosine similarity between raw backbone features therefore compares two spaces that were never trained to be comparable. A matched versus mismatched query test on the development features confirms this: a query's peak relevance on its own audio is statistically indistinguishable from that of a random query, winning only about 53% of the time against a 50% chance level. The backbone space cannot imagine the query. Figure 1 quantifies the consequence and our fix: re-projecting both modalities into the joint space roughly triples Recall1@0.7.

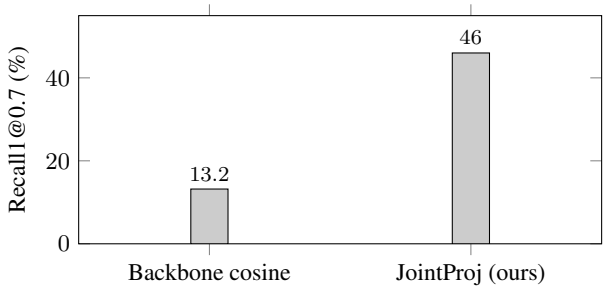

Figure 1: Restoring the joint space unlocks text-space imagination. Recall1@0.7 on the Clotho-Moment validation split (1661 queries, official scorer) rises from 13.2 for raw 768-dimensional backbone cosine to 46.0 for our primary system, which projects both modalities into MS-CLAP's 1024-dimensional joint space (rank-AUC 0.564 to 0.787). Both bars use the same FWHM peak localizers.

Our contributions are: (i) a diagnosis that the provided backbone space is not query-discriminative; (ii) **JointProj**, which reuses MS-CLAP's own projection heads to recover the joint space and a strong per-second relevance curve; and (iii) a careful, ground-truth-validated study on the development data, including a comparison of FWHM and language-model localization [4].

2. METHOD AND EXPERIMENTS

2.1. JointProj: imagining the query in the joint space

For each one-second window we apply MS-CLAP's `audio_encoder.projection` ($768 \to 1024$) to the provided feature and ℓ_2-normalize, giving joint-space audio embeddings $\hat{a}_t$. For the query we re-encode the raw text through the full caption encoder (i.e., `caption_encoder.projection`) to an ℓ_2-normalized vector $\hat{q} \in \mathbb{R}^{1024}$ in Audio-Flamingo-Next-8B [5]. The per-second relevance (saliency) is $s_t = \langle \hat{a}_t, \hat{q} \rangle$, standardized over time. We perform no task-specific training; MS-CLAP is used as a frozen feature extractor, and the projection heads are its own pretrained, contrastively trained weights. A *backbone* baseline computes the same with the raw 768-dimensional features and no projection.

*DCASE 2026 task-6 submission. Work performed at NVIDIA.

Table 1: Development results in the official five-metric format (Recall1 and mAP, higher is better). Top: published QD-DETR baseline on the Castella development-testing split. Bottom: our systems on the Clotho-Moment (CM) validation split (1661 queries). All JointProj (J-Proj) rows share rank-AUC 0.787 and backbone 0.564.

Model	R1@.5	R1@.7	mAP	mAP@.5	mAP@.75
QD-DETR baseline, Castella development-testing (reference)					
Castella only	23.16	10.32	9.11	20.34	6.96
Castella + CM	25.61	13.59	12.06	23.60	10.72
Ours, Clotho-Moment validation					
clap-cosine	19.69	13.18	12.41	21.48	11.72
J-Proj$_{\texttt{sk=5}}$ (s-1)	58.46	46.00	42.52	61.80	45.31
J-Proj$_{\texttt{sk=7}}$ (s-2)	**62.01**	**48.59**	**43.83**	**65.01**	**46.79**
J-Proj$_{\texttt{tap}}$ (s-3)	50.27	38.89	36.96	52.80	38.09

2.2. Localization

We lightly smooth s_t, detect prominent peaks, and emit each peak's FWHM interval $[t_{\text{start}}, t_{\text{end}}]$ ranked by peak height (the top window for Recall1 and up to ten windows for mAP). Boundaries are integer seconds on the 1 Hz grid. The right window width depends on the moment length, and on the development data no single width is best across short and medium moments (Sec. 2.3). We therefore submit three FWHM variants that differ only in the smoothing width `smooth_k` and the relative peak height `rel_height` (a lower value gives tighter windows): system 1 uses `smooth_k=5` (our strongest overall), system 2 uses `smooth_k=7` (favoring medium moments), and system 3 uses `rel_height=0.45` (tighter windows for short moments). Each is the development optimum in a different regime. As a fourth, exploratory submission we include a *novel* language-model localizer: a large language model reads the same joint-space saliency curve and reasons in natural language to place the windows with language-instruction based scoring in the task-activating prompting (TAP) [4]. Its comparison with the deterministic detector is itself informative (Sec. 2.3).

2.3. Development results

Table 1 reports development results. For reference we include the published QD-DETR [6] baseline on the CASTELLA development-testing split as quoted by the task [1]. Our own systems are scored on the Clotho-Moment validation split, the development split for which CLAP features were available to us, with the official scorer. The two splits are not directly comparable, so we read the baseline rows as context and rely on the same-split contrast between backbone cosine and JointProj to isolate our contribution.

We summarize the main findings. First, perception rather than localization is the dominant factor. Projecting into the joint space raises rank-AUC from 0.564 to 0.787 and roughly triples Recall1@0.7 over the backbone at every width, because the backbone space lacks query-specific signal. Second, window width matters and trades off with moment length on the development data, which is why we submit several widths and let the development set choose rather than fixing one. Third, we validate every choice on ground truth: a duration-prior dynamic-programming localizer appeared strong under some settings but was not robust across moment lengths, so we kept FWHM peaks. Fourth, our fourth, exploratory system lets a language model read the joint-space curve and place windows by natural-language reasoning, a genuinely different paradigm, yet it reaches only 16.7 Recall1@0.7 on a development subset, well below the deterministic detector. Inspection shows the language model tends to over-fragment and to mis-set boundary tightness for the strict IoU$\geq$ 0.7 criterion even when it identifies the correct region. The lesson is that, once perception is fixed, the remaining bottleneck is boundary calibration rather than semantic reasoning, a useful negative result for prompting-based localization. Guided by these development results, we submit the four systems above and select the strongest development configuration for the blind evaluation set.

Reproducibility. JointProj applies two frozen MS-CLAP projection modules and the frozen text encoder, and the localization is with Jax, NumPy and SciPy.

3. CONCLUSION

We approached audio moment retrieval as text-space imagination of audio, where a query is located in time by scoring each second of audio against the text in a shared embedding space. JointProj addresses this by passing the provided features and a re-encoded query through MS-CLAP's own projection heads into the joint space, which recovers the cross-modal alignment and produces a sharp per-second relevance curve. On the Clotho-Moment validation split this raised Recall1@0.7 from 13.2 to 46.0 and mAP from 12.4 to 42.5 over the backbone-cosine baseline, with the gain coming from perception rather than from localization. We also found that simple FWHM peak detection is a strong localizer whose ideal window width depends on moment length, and that a language model reasoning over the same curve localizes less accurately because of boundary miscalibration.

4. REFERENCES

[1] H. Munakata, T. Nishimura, S. Nakada, and T. Komatsu, "Language-based audio moment retrieval," in *Proc. IEEE ICASSP*, 2025.

[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)*. IEEE, 2026, pp. 15 352–15 356.

[3] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "CLAP: Learning audio concepts from natural language supervision," in *Proc. IEEE ICASSP*, 2023.

[4] C.-H. H. Yang, Y. Gu, Y.-C. Liu, S. Ghosh, I. Bulyko, and A. Stolcke, "Generative speech recognition error correction with large language models and task-activating prompting," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[5] S. Ghosh, A. Goel, K. Jayakumar, L. Koroshinadze, N. Anand, Z. Kong, S. Gururani, S.-g. Lee, J. Kim, A. Aljafari, *et al.*, "Audio flamingo next: Next-generation open audio-language models for speech, sound, and music," *arXiv preprint arXiv:2604.10905*, 2026.

[6] W. Moon, S. Hyun, S. Park, D. Park, and J.-P. Heo, "Query-dependent video representation for moment retrieval and highlight detection," in *Proc. IEEE/CVF CVPR*, 2023.