CP-JKU SUBMISSION TO TASK 1 OF THE DCASE 2026 CHALLENGE: LLM PREDICTION FUSION AND PSEUDO-LABEL TRAINING FOR HETEROGENEOUS AUDIO CLASSIFICATION

Technical Report

Paul Primus and Gerhard Widmer

Institute of Computational Perception
Johannes Kepler University
Linz, Austria
{paul.primus, gerhard.widmer}@jku.at

ABSTRACT

This report describes our submission to DCASE 2026 Task 1, Heterogeneous Audio Classification. The task requires classifying Freesound audio clips into 23 second-level Broad Sound Taxonomy classes using audio and optional textual metadata. Our system combines pretrained audio-only and audio-text backbones, including PaSST, BEATs, M2D, CP-CLAP, and LAION-CLAP. We use GPT-5.4-mini in two ways: first, to generate metadata summaries for the CLAP text encoders, and second, to estimate class-probability priors from title, tags, and description. These priors are converted into learned class-embedding mixtures and late fused with the backbone representation. To exploit the noisy BSD35k-CS dataset, we use a three-stage training procedure: clean-data training on BSD10k, pseudo-label-based pretraining on BSD35k-CS, and final fine-tuning on BSD10k. The submitted systems are ensembles of selected Stage-3 models and achieve up to 0.834 hierarchical F-score on our BSD10k test split.

***Index Terms*—** DCASE 2026, heterogeneous audio classification, Broad Sound Taxonomy, pseudo-labeling, metadata fusion, CLAP

1. INTRODUCTION AND TASK DESCRIPTION

DCASE 2026 Task 1 addresses heterogeneous audio classification with the two-level Broad Sound Taxonomy (BST) [1, 2]. Systems classify audio clips into one of 23 second-level sound categories. The development data consists of the carefully curated, cleanly annotated BSD10k dataset and the larger BSD35k-CS dataset with noisy annotations, both derived from Freesound audio clips [3, 2, 4]. The task also provides textual metadata, including title, tags, and description, which can be used by multimodal classification systems. The primary evaluation metric is macro-averaged hierarchical F-score [5], which gives partial credit to predictions that are correct at the top level of the taxonomy but incorrect at the second level. More details on the task setup, data, and evaluation protocol are provided on the official task website.[1]

Our submission focuses on two aspects of the task: exploiting textual metadata and using the noisy BSD35k-CS dataset.

- To exploit the metadata, we derive a class-probability prior using GPT-5.4-mini and fuse it with backbone embeddings using a learned class-embedding representation (Section 3).
- To deal with the noisy annotations in BSD35k-CS, we use pseudo-labels produced from Stage-1 predictions, followed by pretraining with those pseudo labels and fine-tuning on the clean BSD10k set (Section 4).

2. SYSTEM OVERVIEW

We train models based on five pretrained backbones. Three are audio-only backbones: PaSST [6], BEATs [7], and M2D [8]. Two are audio-text backbones: LAION-CLAP [9] with a tiny-HTSAT audio encoder [10] and a RoBERTa [11] text encoder, and CP-CLAP [12], a CLAP model based on PaSST and RoBERTa. For the audio-text models, the text input is a GPT-5.4-mini [13] summary generated from the available metadata fields. The prompt used to summarize the metadata is provided in our repository.[2]

3. METADATA PROCESSING AND LLM PREDICTION FUSION

The task metadata can contain useful semantic information, but it is heterogeneous and noisy. In many cases, the metadata describes the acoustic scene or sound source more directly than the audio signal alone, but the relevant evidence may be distributed across title, tags, and free-form description. Large language models are well suited to this setting because they can aggregate these heterogeneous fields, map synonyms and related concepts to the task taxonomy, and produce a compact class-level prior.

We therefore use GPT-5.4-mini [13] to estimate the most likely BST classes from the metadata. The model is prompted with the task class list and the metadata fields and returns a JSON object of the form

$$\{c_1 : p_1, \ldots, c_K : p_K\}, \tag{1}$$

where c_i is a predicted BST class and p_i is its probability. The LLM is also allowed to predict the class "other". However, "other" is not a target class and is therefore not used to weight the class

[1] https://dcase.community/challenge2026/task-heterogeneous-audio-classification

[2] https://github.com/OptimusPrimus/dcase2026_task1

embeddings. We exclude this entry before fusion and normalize the remaining class probabilities to sum to one.

Let $h \in \mathbb{R}^D$ denote the representation produced by the backbone. For audio-only models, h is the pooled audio representation. For CLAP-based models, h is the concatenation of the unnormalized audio and text embeddings, where the text embedding is computed from the GPT-generated metadata summary (as described in Section 2). Let $q \in \mathbb{R}^C$ denote the normalized LLM class-prior distribution over the C target classes. We learn a class-embedding matrix $E \in \mathbb{R}^{C \times D}$ and compute an LLM-derived representation

$$m = qE. \tag{2}$$

The LLM-derived representation m is concatenated with the backbone representation h:

$$z = [h; m]. \tag{3}$$

The concatenated representation is then processed by a fully connected fusion network and a final linear classifier:

$$\hat{y} = Wf(z) + b. \tag{4}$$

The fusion network f is a two-layer fully-connected network with GELU activation and dropout.

4. PSEUDO-LABEL TRAINING WITH BSD35K-CS

To exploit the larger but noisier BSD35k-CS dataset, we use a three-stage training strategy.

4.1. Stage 1: clean-data training

In the first stage, we train multiple multimodal models on BSD10k using the provided ground-truth labels.

Pseudo-labels are generated based on the Stage-1 predictions and then reused in the later stages. For each item, we average the logits from the selected Stage-1 models and convert the result into a probability distribution using softmax:

$$\tilde{y} = \mathrm{softmax}\left(\frac{1}{M}\sum_{m=1}^{M} s_m\right), \tag{5}$$

where s_m denotes the logit vector predicted by model m and M is the number of models in the ensemble.

4.2. Stage 2: noisy-data pretraining

In the second stage, models are trained on BSD35k-CS using both the original noisy labels and the ensemble pseudo-labels. The loss is a linear combination of the cross-entropy loss with the noisy label and a soft cross-entropy loss with the pseudo-label distribution:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{\text{noisy}} + \lambda\mathcal{L}_{\text{pseudo}}. \tag{6}$$

The pseudo-label weight λ is sampled from $[0.75, 1.0]$. We observed that higher values of λ generally led to better validation performance, indicating that the ensemble pseudo-labels were more reliable than the original crowd-sourced labels for many training examples.

4.3. Stage 3: clean-data fine-tuning

In the final stage, models are fine-tuned on BSD10k. We use the same loss structure as in Stage 2, combining the true BSD10k labels with the pseudo-labels generated by the Stage-1 ensemble. The pseudo-label weight is sampled from $[0.95, 1.0]$.

Table 1: Hyperparameter search ranges. Learning rates are sampled log-uniformly; pseudo-label weights λ are sampled uniformly.

Backbone	Stage	LR range	Epochs / λ
PaSST	1	$5 \cdot 10^{-6}$–$1 \cdot 10^{-5}$	30 / –
	2	$3 \cdot 10^{-6}$–$6 \cdot 10^{-6}$	15 / .75–1.0
	3	$0.9 \cdot 10^{-6}$–$2 \cdot 10^{-6}$	15 / .95–1.0
BEATs	1	$3 \cdot 10^{-6}$–$6 \cdot 10^{-6}$	15 / –
	2	$3 \cdot 10^{-6}$–$6 \cdot 10^{-6}$	15 / .75–1.0
	3	$1 \cdot 10^{-6}$–$5 \cdot 10^{-6}$	15 / .95–1.0
M2D	1	$1 \cdot 10^{-6}$–$4 \cdot 10^{-6}$	30 / –
	2	$1 \cdot 10^{-6}$–$3 \cdot 10^{-6}$	30 / .75–1.0
	3	$1 \cdot 10^{-6}$–$3 \cdot 10^{-6}$	15 / .95–1.0
CP-CLAP	1	$1 \cdot 10^{-6}$–$5 \cdot 10^{-6}$	20 / –
	2	$2 \cdot 10^{-6}$–$5 \cdot 10^{-6}$	15 / .75–1.0
	3	$2 \cdot 10^{-6}$–$5 \cdot 10^{-6}$	15 / .95–1.0
LAION-CLAP	1	$1 \cdot 10^{-6}$–$5 \cdot 10^{-6}$	20 / –
	2	$2 \cdot 10^{-6}$–$5 \cdot 10^{-6}$	15 / .75–1.0
	3	$1 \cdot 10^{-6}$–$4 \cdot 10^{-6}$	15 / .95–1.0

5. TRAINING AND MODEL SELECTION

We split BSD10k into training, validation, and test subsets, corresponding to one fold of the task baseline setup. The validation set is used for checkpoint selection, hyperparameter selection, and ensemble construction. The results reported in Section 7 are computed on the held-out BSD10k test split. BSD35k-CS is used only for training in Stage 2 and is not used for validation, model selection, or testing.

Hyperparameters, specifically the learning rate and the pseudo-label loss weight λ, are tuned separately for each backbone and each training stage. The ranges are shown in Table 1. For every backbone-stage combination, we run 20 randomly sampled configurations and select models based on validation-set hierarchical F-score. The batch size is fixed to 18 for all experiments. The learning rate schedule uses linear warmup followed by linear decay to 10^{-7} until the final epoch. We use AdamW [14] with PyTorch's [15] default momentum parameters and fix the weight decay to 0.01. We further use early stopping based on validation hierarchical F-score. Learning rates are sampled log-uniformly, and pseudo-label weights are sampled uniformly. We apply class-frequency loss weighting to counteract class imbalance in BSD10k.

6. SUBMITTED SYSTEMS

We submit four final systems. Each system is an ensemble of selected Stage-3 models. In all systems two CP-CLAP models are used, one CP-CLAP model uses LLM prediction fusion (Section 3) and one does not. The no-fusion CP-CLAP model still uses the RoBERTa text branch with the GPT-generated metadata summary (Section 2) , but it does not use the additional LLM prediction embedding.

- **System 1** combines BEATs, two CP-CLAP models, and one LAION-CLAP model. It uses BEATs, CP-CLAP PaSST, and LAION-CLAP tiny-HTSAT audio representations, CP-CLAP and LAION-CLAP RoBERTa text representations, and LLM

prediction embeddings where enabled.

- **System 2** combines two CP-CLAP models and one M2D model. It uses CP-CLAP PaSST and M2D audio representations, CP-CLAP RoBERTa text representations, and LLM prediction embeddings where enabled. The ensemble has 383M parameters.
- **System 3** uses the same model types and representation types as System 2, but with a different set of trained model instances.
- **System 4** combines two CP-CLAP models, one LAION-CLAP model, and one M2D model. It uses CP-CLAP PaSST, LAION-CLAP tiny-HTSAT, and M2D audio representations, CP-CLAP and LAION-CLAP RoBERTa text representations, and LLM prediction embeddings where enabled.

7. RESULTS

Table 2 reports the performance of the four submitted systems on our custom BSD10k development split, which imitates the official baselines split. System 3 obtains the highest hierarchical F-score on this split.

Table 2: Performance on our BSD10k test split.

System	hP	hR	hF
System 1	0.830	0.831	0.830
System 2	0.831	0.837	0.832
System 3	0.833	0.838	0.834
System 4	0.828	0.832	0.829

8. CONCLUSION

We presented a submission to DCASE 2026 Task 1 based on pretrained audio and audio-text backbones, LLM-based metadata processing, and pseudo-label training for noisily annotated data. The metadata component converts textual metadata into a normalized class prior and fuses it with the backbone representation through learned class embeddings. The training procedure uses ensemble pseudo-labels to exploit BSD35k-CS while reducing the influence of noisy crowd-sourced annotations. Final predictions are obtained by ensembling selected Stage-3 models.

9. ACKNOWLEDGMENT

GPT-5.5 was used to assist in drafting and editing this technical report.

10. REFERENCES

[1] DCASE Community, "DCASE 2026 Challenge Task 1: Heterogeneous Audio Classification," https://dcase.community/challenge2026/task-heterogeneous-audio-classification, 2026, accessed: 2026-06-16.

[2] P. Anastasopoulou, J. Torrey, X. Serra, and F. Font, "Heterogeneous Sound Classification with the Broad Sound Taxonomy and Dataset," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop*, Tokyo, Japan, Oct. 2024.

[3] F. Font, G. Roma, and X. Serra, "Freesound Technical Demo," in *Proceedings of the 21st ACM International Conference on Multimedia*. Association for Computing Machinery, 2013, pp. 411–412.

[4] P. Anastasopoulou and F. Font Corbera, "BSD35k-CS (Broad Sound Dataset 35k - Crowd Sourced)," Mar. 2026. [Online]. Available: https://doi.org/10.5281/zenodo.19187100

[5] P. Anastasopoulou, F. A. Dal Rí, X. Serra, and F. Font, "Hierarchical and Multimodal Learning for Heterogeneous Sound Classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop*, Barcelona, Spain, Oct. 2025.

[6] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," in *Proceedings of Interspeech*, 2022, pp. 2753–2757.

[7] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 5178–5193.

[8] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked Modeling Duo: Learning representations by encouraging both networks to model the input," *arXiv preprint arXiv:2210.14648*, 2022.

[9] Y. Wu, K. Chen, T. Zhang, Y. Hui, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, "Large-Scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2023.

[10] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[11] 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," *arXiv preprint arXiv:1907.11692*, 2019.

[12] P. Primus, F. Schmid, and G. Widmer, "Estimated Audio–Caption Correspondences Improve Language-Based Audio Retrieval," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop*, Tokyo, Japan, Oct. 2024, pp. 121–125.

[13] OpenAI, "GPT-5.4 mini Model Documentation," https://developers.openai.com/api/docs/models/gpt-5.4-mini, 2026, accessed: 2026-06-16.

[14] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.