Adapting Label-Free Concept Bottleneck Models to Audio

Amine Maazizi
ENSTA Paris
MVA Master, ENS Paris-Saclay
amine.maazizi@ensta.fr

Adam Gassem
ENSTA Paris
MVA Master, ENS Paris-Saclay
adam.gassem@ensta.fr

April 2026

Abstract

Concept Bottleneck Models (CBMs) improve interpretability by forcing predictions to pass through human-understandable intermediate concepts. Their main limitation is that they usually require explicit concept annotations, which are expensive to collect and often unavailable. Label-Free Concept Bottleneck Models (LF-CBM) address this limitation by replacing concept supervision with a concept activation matrix extracted from a vision–language foundation model, then learning a bottleneck projection that aligns backbone features with those concept activations [1]. In the original formulation, this mechanism is designed for images and relies on CLIP.

In this paper, we adapt LF-CBM to audio classification by replacing the image–text similarity stage with an audio–text similarity stage based on CLAP [2]. We keep the LF-CBM structure unchanged: a strong backbone produces features, CLAP defines concept supervision through a concept activation matrix, and a sparse classifier predicts labels from the learned bottleneck. We instantiate the pipeline with AST[3] as the audio backbone and evaluate on three datasets with different difficulty profiles: ESC-50, UrbanSound8K, and CREMA-D. This setup allows us to compare predictive performance before and after bottlenecking and to analyze where label-free concept constraints preserve or reduce accuracy in practice. The code for the Label-free CBM audio pipeline is available on GitHub: https://github.com/Adam-Ousse/Label-free-CBM-Audio.

1 Introduction

Modern deep neural networks achieve strong performance on perception tasks, but their decision process is often difficult to understand. In audio, this problem is particularly frustrating because many tasks are naturally described in semantic terms: a clip may contain *dog barking*, *rain falling*, *engine noise*, or *crowd cheering*, yet a standard classifier represents these signals with uninterpretable latent features. The resulting gap between predictive success and semantic understanding motivates methods that aim to produce faithful and human-readable intermediate representations.

Concept Bottleneck Models are a particularly appealing approach to this problem. Instead of directly predicting the label from the input, a CBM first predicts interpretable concepts and only then uses those concepts to make the final decision. This structure gives the user access to a semantically meaningful intermediate layer and makes it possible to inspect or intervene on the reasoning process. However, classical CBMs require concept annotations during training, and this requirement is often the main obstacle to their adoption. For many datasets, especially outside highly curated domains, concept labels simply do not exist at scale.

Label-Free Concept Bottleneck Models[1] aims to overcome this difficulty. The key insight of LF-CBM is that one can supervise the bottleneck layer without explicit concept labels by using a foundation model as a source of semantic alignment. In the original paper, CLIP is used to compute

a concept activation matrix over the training set, and a projection is learned so that the bottleneck neurons align with those concept activations. The method preserves the interpretability advantages of CBMs while avoiding manual concept annotation, and it scales to large vision datasets.

We study whether LF-CBM can transfer to *audio*. This is not a simple modality swap: relevant cues are spread over time, events can overlap, and short informative sounds are often mixed with background noise or silence. Weak labels further increase ambiguity. Still, audio-language models offer a direct link between waveform content and text concepts, which makes this transfer testable in practice.

The main contribution of this paper is therefore methodological: we explain how the LF-CBM pipeline can be adapted from image classification to audio classification by replacing CLIP with CLAP and by redesigning the concept generation and training pipeline accordingly. We evaluate this adaptation on three datasets that cover different audio settings: ESC-50[4], UrbanSound8K[5], and CREMA-D[6].

2 Label-Free CBM

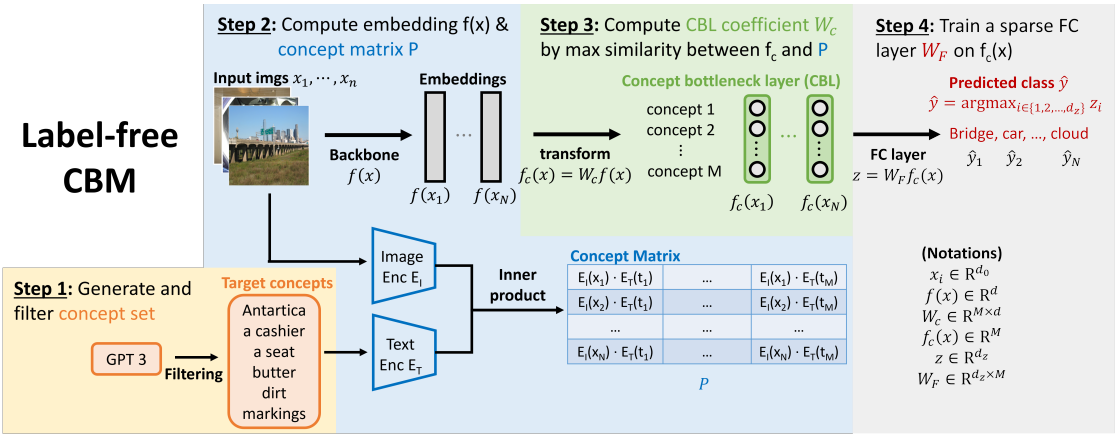


Figure 1: Overview of paper's pipeline for creating label-free CBM[1]

2.1 Motivation

The original LF-CBM paper starts from the observation that standard CBMs are limited by two major issues: they require labeled concept data, and they often lose a substantial amount of accuracy compared to standard deep networks [1]. LF-CBM addresses both issues by combining a pretrained backbone with a foundation model that provides concept-level supervision indirectly. The method is post-hoc in spirit: instead of training an interpretable model from scratch, it takes a strong backbone and transforms it into a concept bottleneck model after the fact.

This construction has three attractive properties. First, it is *label-free* with respect to concepts: no *manual* concept annotations are needed. Second, it is *scalable*: the original paper demonstrates the approach on datasets up to ImageNet scale. Third, it is *modular*: the backbone and the concept supervision source play distinct roles, which makes the framework adaptable to any deep learning model [1].

2.2 Concept Set Creation and Filtering

The first step of LF-CBM is to create a set of candidate concepts. In the original paper, this is done automatically using GPT-3 prompts that ask for important visual features, common surroundings, and superclasses for each target class [1]. The goal is to avoid relying on hand-crafted expert concept inventories and instead generate a concept vocabulary automatically from class names.

Because the raw concept set is noisy, a series of filters is then applied. Concepts that are too long are removed in order to keep the vocabulary simple and readable. Concepts too similar to class names are removed to prevent trivial explanations. Near-duplicate concepts are also removed to avoid redundancy. Finally, concepts that do not meaningfully activate on the training data or cannot be projected faithfully into the bottleneck are discarded [1]. The important point is that concept filtering is not merely cosmetic: it directly affects interpretability, computational cost, and the stability of the learned bottleneck.

2.3 Concept Activation Matrix

Once the concept set is fixed, LF-CBM computes a concept activation matrix on the training set. Let the training data be $D = \{x_1, \dots, x_N\}$ and the concept set be $C = \{t_1, \dots, t_M\}$. In the original vision formulation, CLIP provides image embeddings $E_I(x_i)$ and text embeddings $E_T(t_j)$, and the concept activation matrix is defined by

$$P_{i,j} = E_I(x_i)^\top E_T(t_j).$$

Thus, $P \in \mathbb{R}^{N \times M}$ records how strongly each training image aligns with each textual concept [1]. This matrix is not the output of the backbone itself. Rather, it acts as a semantic target that the future bottleneck layer should emulate.

This separation is one of the most important structural ideas in LF-CBM. The backbone remains a task-oriented feature extractor, while the foundation model provides a semantic coordinate system over concepts. The bottleneck layer is then learned as a bridge between the two.

2.4 Learning the Concept Bottleneck Layer

Let $f(x) \in \mathbb{R}^{d_0}$ denote the backbone representation. LF-CBM learns a projection matrix $W_c \in \mathbb{R}^{M \times d_0}$ and defines the concept bottleneck representation

$$f_c(x) = W_c f(x).$$

The purpose of this projection is to make each coordinate of $f_c(x)$ correspond to a target concept. To do this, LF-CBM compares the activation pattern of each bottleneck neuron across the training set with the corresponding column of the concept activation matrix P [1].

The original paper introduces a differentiable similarity objective, based on a normalized cosine-style similarity after cubing activations, in order to emphasize highly activating examples. The projection is optimized so that bottleneck activations correlate with the target concept activations. Concepts whose learned neuron does not align sufficiently well with the target are then removed. As a result, the final bottleneck is not merely a fixed list of candidate concepts: it is a filtered set of concepts that can actually be represented in the backbone feature space.

2.5 Sparse Final Layer

After the bottleneck has been learned, LF-CBM fits a sparse linear classifier on top of it. If the number of classes is d_z, the final layer is a matrix $W_F \in \mathbb{R}^{d_z \times M}$ and a bias vector b_F, producing

class logits

$$z(x) = W_F f_c(x) + b_F.$$

Sparsity is enforced with an elastic-net style objective, so that each output class depends on only a limited number of concepts [1]. This final sparsity is central to the interpretability of the model: without it, the decision could spread over too many concepts to remain understandable. The resulting classifier can then be interpreted globally through class-specific concept weights and locally through concept contribution scores on a given input.

3 Adaptation to Audio

4 Label-Free CBM

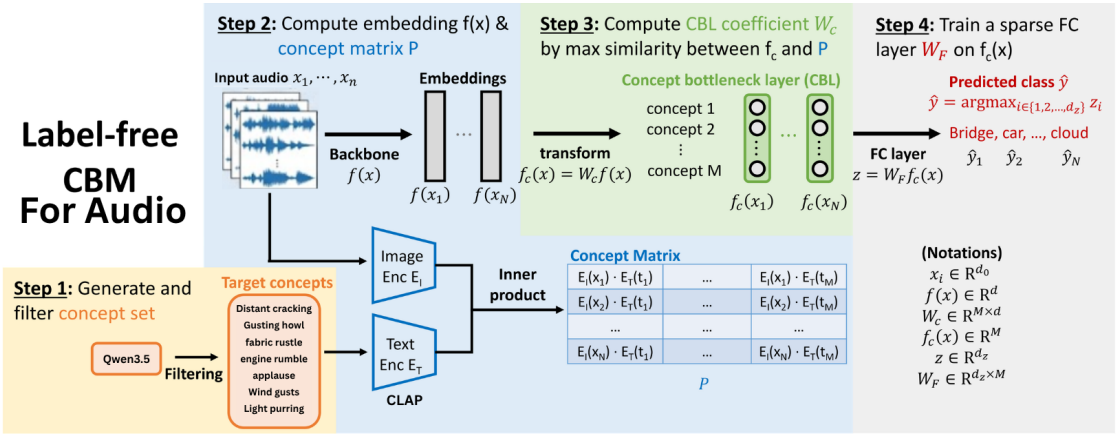


Figure 2: Overview of pipeline for creating label-free CBM for Audio

4.1 From Image–Text to Audio–Text Similarity

The core idea of our adaptation is simple: we preserve the LF-CBM pipeline, but replace the image–text foundation-model stage with an audio–text stage. In the original method, the concept activation matrix is computed with CLIP. In our method, it is computed with CLAP.

CLAP learns a joint audio–text space by training an audio encoder and a text encoder with a contrastive objective over paired audio and text data [2]. At inference time, CLAP supports zero-shot classification by comparing audio embeddings and text label embeddings in this shared space. This makes it a natural audio analogue of CLIP and therefore a natural replacement in LF-CBM.

Formally, let E_a denote the CLAP audio encoder and E_t denote the CLAP text encoder. Given audio clips $\{x_i\}_{i=1}^N$ and textual concepts $\{c_j\}_{j=1}^M$, we define the audio concept activation matrix

$$P_{i,j} = E_a(x_i)^\top E_t(c_j).$$

Equivalently, in matrix form,

$$P = E_a(X)\, E_t(C)^\top.$$

This matrix plays exactly the same role as the CLIP-derived matrix in the original LF-CBM: it provides concept-level targets over the training set, without requiring concept labels.

4.2 Audio-Oriented Concept Sets

Adapting LF-CBM to audio, in our case, was mainly a *prompt-and-filter* pipeline rather than a manual redesign of concept types. We did not hand-build a taxonomy of spectral or temporal descriptors. Instead, for each class, we generated candidate concepts with three fixed prompt families using Qwen3.5-9B[7] and then filtered them aggressively before training. For details on the complete set of prompts used, refer to B.

- **important**: discriminative audible cues for the target class,
- **around**: sounds commonly co-occurring with the class (excluding the class itself),
- **superclass**: broader sound-type categories.

This gave a large raw concept pool, which we then cleaned by removing duplicates/near-duplicates, label-leakage terms too close to class names, and overly generic or weakly informative concepts. During LF-CBM training, concepts were further pruned by the standard activation/projection faithfulness checks. So the practical adaptation was: *audio-specific prompting + automatic filtering + LF-CBM pruning*, not manual concept engineering.

4.3 Choice of Audio Backbone

For the audio backbone, we consider models that produce stable clip-level features and are compatible with broad environmental sound classification. Among the most principled candidates is AST. AST[3] is a pure-attention audio classifier operating on spectrogram patches and has reported strong performance on both AudioSet[8] and ESC-50 [4]. In particular, AST reaches 0.485 mAP on AudioSet and 95.6% accuracy on ESC-50 in its original evaluation [3]. This makes it an attractive backbone for our adaptation.

AST is especially appealing in our setting for three reasons. First, it provides a strong audio representation. Second, it works naturally on time–frequency inputs, which remain a convenient substrate for interpretation. Third, it is flexible across datasets of different durations [3]. Our adaptation therefore uses AST as a reference backbone candidate, while remaining open to testing other pretrained audio encoders if needed.

4.4 Expected Challenges in Audio

Although the replacement CLIP $\rightarrow$ CLAP is structurally natural, the audio setting introduces challenges that are more severe than in vision. Audio events are often temporally sparse, and long silent regions or irrelevant background sound can dominate a clip. Different events may overlap in time, and the same sound class may exhibit strong acoustic variability. Moreover, weakly labeled web-scale datasets can contain clips whose metadata does not fully describe the salient acoustic content.

This is one reason why we do not frame the project mainly around a narrow bird-chirp challenge. Such a challenge may be useful as an application case, but it contains limited event diversity and many low-information segments. It is therefore not ideal as the primary benchmark for concept discovery. Broader environmental sound datasets are better suited to testing whether label-free concept bottlenecks can recover meaningful and reusable semantic structure.

5 Training Plan and Experimental Setup

5.1 Datasets and Split Protocols

We evaluate on three datasets that cover different audio settings:

1. **ESC-50**[4]: 2,000 clips, 50 classes. The AST baseline is reported from the established ESC-50 evaluation protocol (five-fold setting). The CBM result is reported on the fixed split used in our LF-CBM pipeline outputs: `fold1_train`, `fold1_val`, `fold1_test`.
2. **UrbanSound8K**[5]: 8,732 clips, 10 classes. We follow the official fold protocol with folds 1–8 for training, fold 9 for validation, and fold 10 for final test.
3. **CREMA-D**[6]: speech emotion recognition with 6 classes (anger, disgust, fear, happy, neutral, sad). We keep the official `test.csv` as test, and split the official `train.csv` into train/val with a stratified 90/10 partition.

5.2 Training and CBM Pipeline

Each dataset follows the same high-level workflow:

1. Fine-tune an AST classifier as the predictive baseline.
2. Build dataset manifests and class mappings in a unified format.
3. Generate and filter concept candidates from class names and audio-oriented prompts.
4. Encode concepts and clips with CLAP to form the concept activation matrix P.
5. Learn the LF-CBM projection matrix W_c from AST features to concept space.
6. Train the sparse final linear layer on top of the concept bottleneck.
7. Evaluate baseline AST and LF-CBM on the same held-out test split.

This setup isolates the effect of bottlenecking because AST is the shared backbone before and after conversion to LF-CBM.

The fine-tuned models, which were trained using different datasets, have been uploaded to Hugging Face for easy access and further experimentation:

- Adam-ousse/ast-esc50-finetuned-fold1
- Adam-ousse/ast-urbansound8k-finetuned-fold10
- Adam-ousse/ast-cremad-finetuned

5.3 Implementation Notes

Audio is standardized at 16 kHz in the data pipeline and represented through explicit train/val/test manifests for reproducibility. For CREMA-D, we apply class-weighted cross-entropy during AST fine-tuning to mitigate class imbalance. The LF-CBM stage uses a sparse elastic-net style final layer to preserve interpretability through compact concept support per class.

6 Results

Table 1: AST baseline (before bottleneck) and LF-CBM (after bottleneck) on each dataset.

Dataset	AST Acc. (%)	AST Loss	CBM Acc. (%)	CBM Loss	ΔAcc. (pts)
ESC-50	93.50	0.2708	94.00	0.2819	+0.50
UrbanSound8K	89.01	0.5032	87.57	0.3806	-1.43
CREMA-D	70.05	1.7674	67.36	0.9616	-2.69

6.1 Quantitative Summary

Table 1 shows three distinct regimes. On ESC-50, LF-CBM slightly improves accuracy (+0.50 points) with almost unchanged loss. On UrbanSound8K and CREMA-D, LF-CBM remains competitive but trails AST by 1.43 and 2.69 points respectively. This pattern suggests that bottleneck constraints are easiest to preserve on cleaner environmental benchmarks and harder on datasets with higher class overlap or stronger imbalance.

The loss values should be interpreted with care because training objectives differ across stages. For example, CREMA-D AST uses class-weighted cross-entropy, while the CBM classifier optimizes the sparse linear stage on concept activations. Accuracy remains the primary metric for model-level comparison.

6.2 Qualitative Analysis

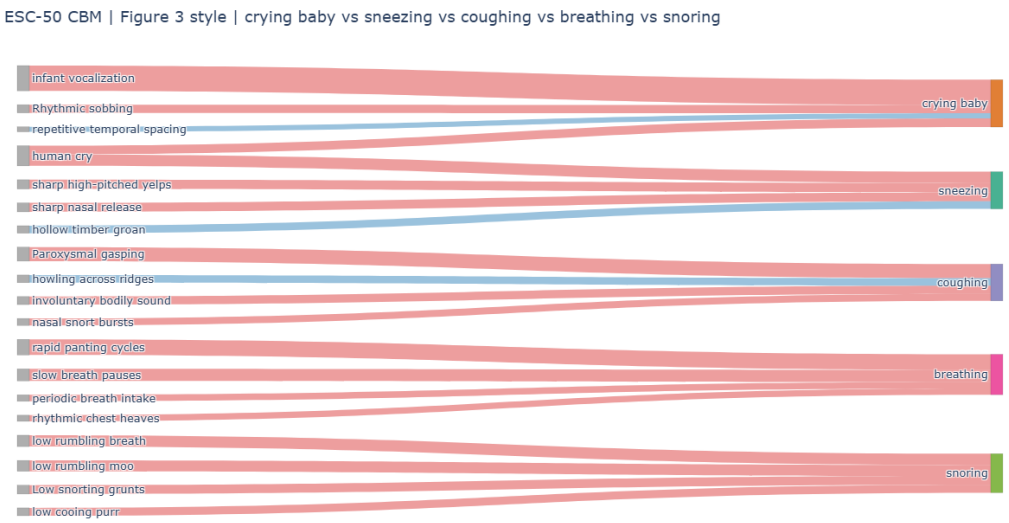
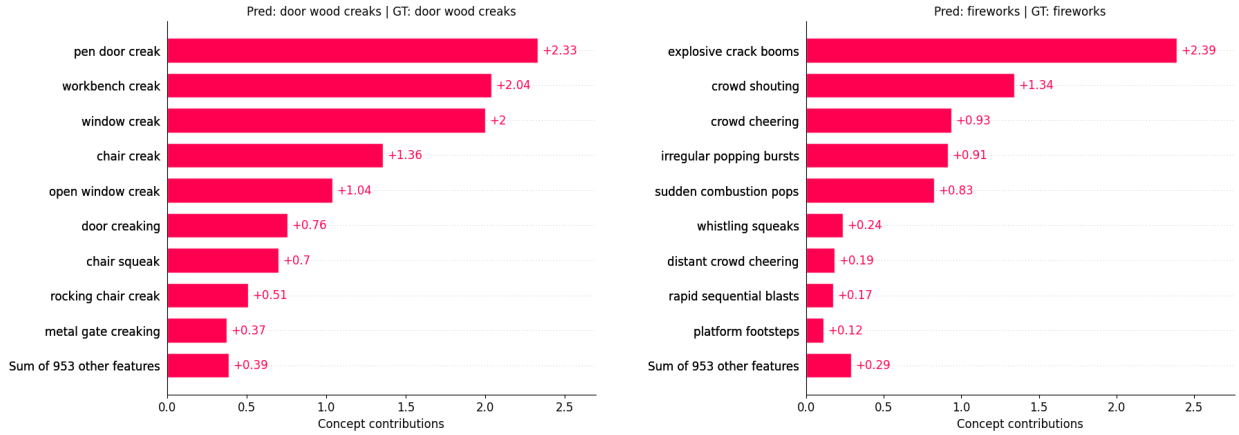
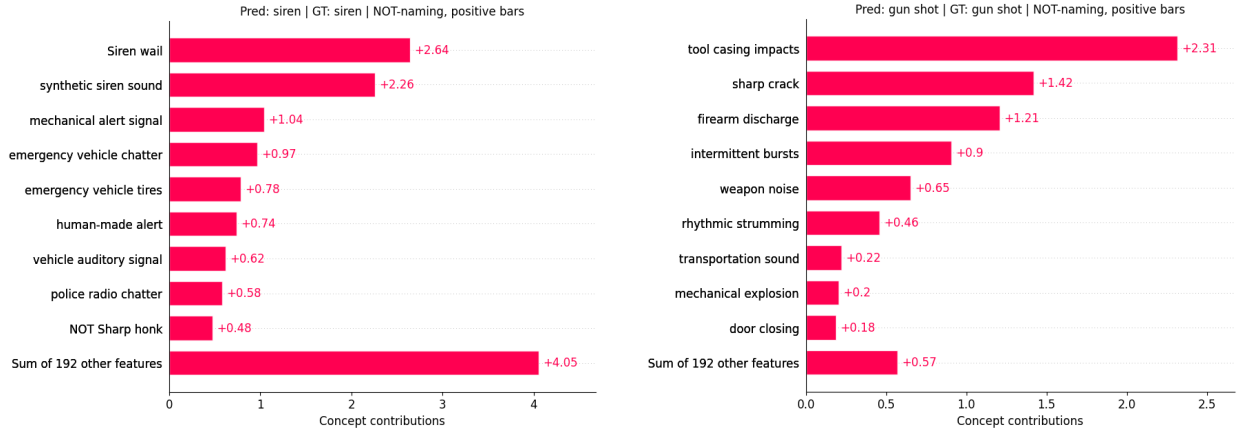


Figure 3: Visualization of the final layer weights of our Label-free CBM. Showcasing how the model differentiate between similar classes.



(a) door wood creaks (b) fireworks

Figure 4: ESC-50 concept-level explanation examples (1 row, 2 columns).



(a) siren (b) gunshot

Figure 5: UrbanSound8K concept-level explanation examples (1 row, 2 columns).

For more qualitative results and to listen to the audio, we recommend checking the GitHub pages, where we have created a web interface with both the audio and the results: https://github.com/Adam-Ousse/Label-free-CBM-Audio.

6.3 ESC-50 Model Selection and Main Ablations

Our ESC-50 experiments serve two purposes. First, they validate the audio LF-CBM pipeline end-to-end on a controlled benchmark. Second, they isolate which LF-CBM design choices matter most when moving from image–text alignment to audio–text alignment. In the main text, we focus on the four ablation axes that were most informative for model selection: projection similarity objective, final-layer sparsity, prompt template, and projection-faithfulness threshold. The shared experimental protocol and the secondary ablation axes are reported in the appendix.

Table 2: Main ESC-50 model-selection ablations on the fixed split used during development.

Ablation	Setting	# Retained	Avg. NNZ/Class	Test Acc. (%)
Similarity objective	Cosine	782	55.60	85.25
	Cosine-cubed	782	59.30	**90.25**
Final layer sparsity	Dense	782	781.82	88.50
	Sparse (moderate)	782	59.38	**90.50**
	Sparse (strong)	782	22.72	89.25
Prompt template	Raw[1]	782	60.32	89.75
	Sound-of[2]	783	59.92	**92.00**
	Audio-recording[3]	795	60.74	89.75
Projection threshold	0.35	782	60.12	**90.50**
	0.45	782	59.16	90.25
	0.55	782	59.28	**90.50**

Table 2 yields four main conclusions. First, the projection similarity objective is a major component of the method: replacing plain cosine with cosine-cubed improves test accuracy from 85.25% to 90.25% while increasing average projection similarity from 0.9687 to 0.9916. This is consistent with the original LF-CBM paper, where cosine-cubed is introduced precisely to improve alignment over a simple cosine baseline while remaining differentiable [1]. In our audio setting, this confirms that the projection-learning mechanism itself transfers meaningfully beyond vision.

Second, the sparsity regime of the final layer materially affects generalization. The best performance is obtained with a moderately sparse classifier, which reaches 90.50% test accuracy with about 59.4 nonzero weights per class. The dense head is less accurate at 88.50% despite near-perfect validation performance, suggesting stronger overfitting of the concept classifier on this small benchmark. This differs from the dense-versus-sparse trend reported in the original LF-CBM appendix, where removing sparsity often improves accuracy while reducing interpretability. Here, moderate sparsity appears to help both interpretability and test-time robustness.

Third, prompt wording for CLAP concept encoding is a genuinely audio-specific design choice. The best prompt template is `the sound of {concept}`, which reaches 92.00% test accuracy, compared with 89.75% for both the raw concept string and the more verbose `an audio recording of {concept}` template. Since retained concept counts and classifier sparsity remain nearly unchanged across these runs, the gain is best understood as improved audio–text alignment rather than a simple change in model size.

Fourth, the projection-faithfulness threshold is comparatively stable in the tested range. Varying the cutoff from 0.35 to 0.55 leaves the retained concept count unchanged at 782 and changes test accuracy by only 0.25 points. We therefore keep the original LF-CBM default value of 0.45 for the final configuration, since it remains directly comparable to the original method and performs essentially identically to nearby values.

Overall, the ESC-50 ablations indicate that the most consequential design choices for audio LF-CBM are the projection objective, the text prompt used to encode concepts, and the sparsity regime of the final classifier. Additional methodological details and secondary ablation axes, including concept-set-size and filtering-policy sweeps, are provided in the appendix.

[1] `{concept}`

[2] `the sound of {concept}`

[3] `an audio recording of {concept}`

7 Discussion

The three-dataset comparison highlights a practical trade-off. LF-CBM keeps strong semantic interpretability and remains close to AST on all benchmarks, but absolute accuracy is not uniformly preserved after bottlenecking. The small gain on ESC-50 indicates that concept regularization can help on cleaner datasets. The drops on UrbanSound8K and CREMA-D indicate that concept projection and sparsity can discard some discriminative detail when classes are acoustically confusable.

From an interpretability perspective, this behavior is still useful: the model family remains transparent enough to inspect concept contributions per prediction, and the performance gap is moderate rather than catastrophic. For future work, the main lever is reducing information loss in the projection step, for example through stronger concept filtering, dataset-specific prompt refinements, and mild relaxation of final-layer sparsity on harder datasets.

8 Conclusion

We implemented and evaluated a complete audio LF-CBM pipeline with AST backbones and CLAP-based concept supervision on ESC-50, UrbanSound8K, and CREMA-D. The final comparison confirms that the approach is technically robust across datasets and preserves strong baseline quality after bottlenecking.

The key empirical message is simple: LF-CBM can match or nearly match AST while adding concept-level transparency, but the accuracy-interpretability balance depends on dataset difficulty. In our runs, ESC-50 benefits slightly from bottlenecking, while UrbanSound8K and CREMA-D show moderate drops. This motivates a next iteration focused on improving concept quality and projection fidelity on harder settings.

References

[1] T. Oikarinen, S. Das, L. M. Nguyen, and T.-W. Weng, "Label-free concept bottleneck models," 2023.

[2] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "Clap: Learning audio concepts from natural language supervision," 2022.

[3] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," 2021.

[4] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1015–1018, 2015.

[5] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, (Orlando, FL, USA), pp. 1041–1044, Nov. 2014.

[6] H. Cao, D. Cooper, M. Keutmann, R. Gur, A. Nenkova, and R. Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset." GitHub Repository, 2014. Accessed: 2026-04-17.

[7] Q. Team, "Qwen3.5-9b: A multimodal foundation model." Hugging Face Model Repository, 2026. Accessed: 2026-04-17.

[8] L. Gemma Roig, S. Kasun, *et al.*, "Audioset: A large-scale dataset of manually annotated audio events." Google Research Dataset, 2017. Accessed: 2026-04-17.

A Hyperparameter Sweeps and Ablations

This appendix reports the detailed ESC-50 sweeps used for model selection. In the main text, we retain only the primary ablations needed to motivate the final configuration. Here, we provide the shared experimental protocol and the secondary ablation axes that would otherwise take too much space in the main paper.

A.1 ESC-50 Ablation Protocol

All ESC-50 ablations were run through the same training pipeline and differ by only one targeted variable at a time. Unless explicitly stated otherwise, runs share the same backbone, the same initial concept inventory, the same concept activation cutoff, and the same split configuration recorded in the experiment outputs. Concretely, each run proceeds as follows:

1. optionally truncate the concept inventory before training,
2. remove weak concepts using the CLAP top-activation filter,
3. compute the audio–text concept activation matrix,
4. learn the concept projection layer with the chosen similarity objective,
5. filter concepts using the projection-faithfulness threshold,
6. train the final linear classifier with elastic-net regularization,
7. evaluate on the held-out test split.

This design makes the sweeps controlled: each ablation modifies a single modeling component while reusing the same overall LF-CBM pipeline.

A.2 Secondary Ablation Axes

Beyond the primary ablations discussed in the main text, we also evaluate two additional axes on ESC-50:

1. concept-set size under hard top-K truncation,
2. concept filtering policy.

These sweeps are useful for understanding the stability of the method, but they are secondary to the main model-selection questions because they do not change the core projection-learning mechanism.

A.3 Detailed ESC-50 Tables

Table 3: Projection-objective and final-layer sparsity ablations on ESC-50.

Setting	Val. Similarity	# Retained Concepts	Avg. NNZ/Class	Test Accuracy (%)
Cosine	0.9687	782	55.60	85.25
Cosine-cubed	0.9916	782	59.30	90.25
Dense final layer	0.9916	782	781.82	88.50
Sparse (moderate)	0.9915	782	59.38	90.50
Sparse (strong)	0.9915	782	22.72	89.25

Table 4: Projection-threshold and concept-set-size ablations on ESC-50.

Setting	# Retained Concepts	Avg. NNZ/Class	Test Accuracy (%)
Threshold 0.35	782	60.12	90.50
Threshold 0.45	782	59.16	90.25
Threshold 0.55	782	59.28	90.50
Top-50 concepts	46	25.62	82.25
Top-100 concepts	95	34.72	89.00
Top-200 concepts	193	42.94	87.00

Table 5: Prompt-template and filtering-policy ablations on ESC-50.

Setting	# Retained Concepts	Avg. NNZ/Class	Test Accuracy (%)
`{concept}`	782	60.32	89.75
`the sound of {concept}`	783	59.92	92.00
`an audio recording of {concept}`	795	60.74	89.75
Full filtering	782	60.38	90.25
No class-similarity filter	801	58.82	90.25
No redundancy filter	968	63.34	90.25

B Prompts Used for Concept Creation

B.1 Important Prompt

important

```
You are an expert in sound perception.

Task: List 3-5 of the most important auditory concepts that help recognize the sound class "{class}".

Requirements:
- Concepts must be directly audible (no visual or semantic descriptions)
- Use concise, interpretable phrases (1-3 words)
- Prefer informative multi-word concepts when useful (e.g., "rapid barking", "steady rainfall")
- Single words are allowed if they are clear and meaningful
- Focus on discriminative cues (what makes this class distinct)
- Avoid vague terms (e.g., "noise", "sound")
- Avoid duplicates or synonyms

Examples:

Class: dog
- loud barking
- low growling
- rapid panting
- sharp whining
- distant howling

Class: rain
- steady rainfall
- light dripping
- irregular splashing
- distant thunder
- wind noise
```

```
Now list 3-5 of the most important auditory concepts for "{class}":
```

B.2 Superclass Prompt

superclass

```
You are an expert in sound categorization.

Task: Provide 3-5 of superclass categories for the sound class "{class}".

Requirements:
- Use short, interpretable labels (1-3 words)
- Prefer meaningful phrases over single generic words
- Return 2-5 categories
- Include multiple abstraction levels (specific -> general)
- Categories must describe types of sounds, not contexts
- Avoid redundancy

Examples:

Class: dog
- animal vocalization
- mammal sound
- domestic animal sound
- biological sound source

Class: rain
- weather sound
- natural ambient sound
- environmental soundscape
- atmospheric sound

Now provide 3-5 superclasses for "{class}":
```

B.3 Around Prompt

around

```
You are analyzing real-world audio recordings.

Task: List 3-5 of the sounds that commonly occur together with the sound class "{class}".

Requirements:
- Do NOT include the main class itself
- Use concise, interpretable phrases (1-3 words)
- Prefer informative multi-word concepts (e.g., "distant traffic noise", "human conversation")
- Focus on clearly audible elements in the environment
- Avoid vague terms (e.g., "background noise")
- Avoid duplicates

Examples:

Class: dog
- human speech
- footsteps on ground
- door opening sounds
- leash movement noise
- park ambience

Class: rain
- wind noise
```

```
- distant traffic
- thunder rumble
- flowing water
- umbrella movement

Now list 3-5 sounds commonly heard around "{class}":
```