LEVERAGING PREDICTION ENTROPY FOR AUTOMATIC PROMPT WEIGHTING IN ZERO-SHOT AUDIO-LANGUAGE CLASSIFICATION

Karim El Khoury[†,1] *Maxime Zanella*[†,1,2] *Tiffanie Godelaine*[†,1]
Christophe De Vleeschouwer[1] *Benoît Macq*[1]

[1]ICTEAM, UCLouvain, Belgium [2]ILIA, UMons, Belgium

ABSTRACT

Audio-language models have recently demonstrated strong zero-shot capabilities by leveraging natural-language supervision to classify audio events without labeled training data. Yet, their performance is highly sensitive to the wording of text prompts, with small variations leading to large fluctuations in accuracy. Prior work has mitigated this issue through prompt learning or prompt ensembling. However, these strategies either require annotated data or fail to account for the fact that some prompts may negatively impact performance. In this work, we present an entropy-guided prompt weighting approach that aims to find a robust combination of prompt contributions to maximize prediction confidence. To this end, we formulate a tailored objective function that minimizes prediction entropy to yield new prompt weights, utilizing low-entropy as a proxy for high confidence. Our approach can be applied to individual samples or a batch of audio samples, requiring no additional labels and incurring negligible computational overhead. Experiments on five audio classification datasets covering environmental, urban, and vocal sounds, demonstrate consistent gains compared to classical prompt ensembling methods in a zero-shot setting, with accuracy improvements 5-times larger across the whole benchmark.

Index Terms— Prompt Weighting, Entropy Minimization, Zero-Shot Classification, Audio-Language Models

1. INTRODUCTION

Unlike approaches learning from predefined labels, Audio-Language Models (ALMs) utilize natural language as a supervision signal, which is more suitable for describing complex real-world audio recordings. This progress has been made possible by the development of self-supervised contrastive pre-training methods and the availability of large-scale text-audio datasets [1–3]. Contrastive pre-training on these large scale multimodal datasets is a powerful framework to obtain models that generalize well across diverse audio classification tasks. This design allows one to construct an audio classifier directly from textual descriptions of categories, without the need for additional training. Inspired by advancements in vision-language models such as CLIP [4], CLAP [5, 6] is one of the first works to investigate its use in the audio field [3, 7–9].

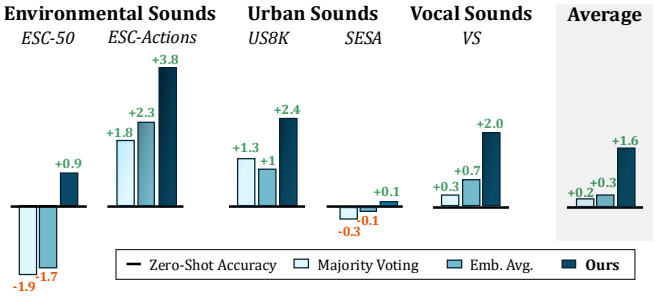

Fig. 1: Summary of classification accuracy improvement over zero-shot prediction. Our approach, applied on the entirety of each dataset, shows consistent improvement over majority voting and embedding averaging ensemble methods.

Despite ALMs having solid zero-shot classification accuracy [5], designing the right prompt for a given class becomes challenging, something even hindering final performance [10]. In fact, a slight change in a prompt can lead to vastly different predictions, potentially making the performance of a model unreliable. One solution to mitigate this problem is to rely on few-shot prompt learning [10]. However, this approach can be computationally intensive and requires annotated samples. Another solution involves augmenting prompts using large language models [11, 12]. While this is an interesting option, it is complementary to our line of work, as such methods can benefit from optimized prompt weighting to improve their overall performance. A third solution is to use an ensemble of templates [5–7, 13] (Table 1 shows common templates in the audio-language literature) to mitigate the sensitivity of models to the specific wording of a single prompt. Since selecting the right prompt among all templates is difficult, majority voting is often used on top of individual prompt predictions to make a final prediction. Another method is to simply use, for each class, the average of the embeddings of its individual prompts. These averaged embeddings are then used for classification.

While averaging text prompts is convenient, it is clear that some prompts perform significantly worse than others (see Figure 2). In a purely zero-shot setting, it is however not straightforward to identify poorly performing prompts. Specifically, we hypothesize that appropriate weighting factors to average the text prompts should induce predictions with high confidence. Hence, we propose an entropy minimization objective, where the goal is to find a weighting vector, β, that aims to create a weighted combination of prompts that results in predictions with low entropy.

[†]The authors have contributed equally to this work.

Acknowledgments – M.Z. is funded by the ARIAC (Walloon region grant No. 2010235). T.G. is funded by MedReSyst (EU-Wallonia 2021-2027 program).

Contributions. We introduce a principled optimization framework that treats prompt ensembling as a problem of finding weights that maximize prediction confidence. This is achieved by minimizing the entropy of the final prediction in a purely zero-shot setting. Our approach allows to handle poorly performing prompts and overcomes the limitations of traditional ensembling methods, delivering consistent accuracy improvements. A summary of our results showing improvements over baseline ensemble methods on five audio classification benchmarks is shown in Figure 1.

2. METHOD

We present our iterative algorithm for finding prompt-weighting vector, β, by minimizing an objective function that drives for low-entropy (high-confidence) predictions. We regularize it against terms that prevent deviation from an initial zero-shot estimate and encourage smooth weight distribution.

2.1. Problem formulation

We start by encoding each audio sample $i \in \{1, \cdots, N_S\}$ with the ALM audio encoder to obtain a feature vector $\mathbf{f}_i \in \mathbb{R}^{1\times d}$. We get a set of text embeddings $\mathbf{t}_{jk} \in \mathbb{R}^{1\times d}$, where $\mathbf{t}_{jk}$ is obtained by combining template $j \in \{1, \cdots, N_T\}$ with class $k \in \{1, \cdots, N_C\}$ (e.g., *This is a sound of* $\{$*class k*$\}$) and passing it into the ALM text encoder. We then compute the logit l_{ijk} of sample i for template j and class k as the cosine similarity*:

$$l_{ijk} = \mathbf{f}_i \mathbf{t}_{jk}^T \tag{1}$$

Next, we compute the average logit $\bar{l}_{ik}$ of sample i for class k as a weighted average of the logits computed across all templates j. This is given by:

$$\bar{l}_{ik} = \sum_{j=1}^{N_T} \beta_j l_{ijk} \tag{2}$$

where $\beta \in \mathbb{R}^{1\times N_T}$ is the weighting vector. In the uniform case, $\beta = \frac{1}{N_T}\mathbf{1}_{N_T}$. By applying the softmax function, we obtain the prediction p_{ik} of sample i for class k:

$$p_{ik} = \frac{\exp(\bar{l}_{ik}/\tau)}{\sum_{l=1}^{N_C}\exp(\bar{l}_{il}/\tau)} \tag{3}$$

where τ is a temperature scaling hyperparameter that controls the smoothness of the predictions.

In the case where β is uniform, each template contributes equally to the final logit. However, we hypothesize that this method is sub-optimal, particularly when the classification accuracy across templates varies greatly, as observed in Figure 2. Hence, we aim to find the optimal contribution of each template, so that the model outputs confident predictions, i.e. low entropy p_i vectors, across an entire batch of data samples.

> We formulate our optimization objective as finding a weighting vector β that produces confident predictions on average.

*Note that all vectors are normalized to the unit hypersphere, therefore the cosine similarity reduces to the dot product.

2.2. Objective variable

Let the variables $\beta \in \mathbb{R}^{1\times N_T}$ represent the weights for each of the N_T distinct text prompt templates. This vector is constrained such that it lies in the probability simplex, enforcing the following conditions: $\sum_{j=1}^{N_T}\beta_j = 1$, and $\beta_j \geq 0 \quad \forall j$.

2.3. Objective function

The goal is to minimize the objective function $\mathcal{L}(\beta)$ composed of three terms:

$$\mathcal{L}(\beta) = \frac{1}{N_S}\sum_{i=1}^{N_S}\Big(\underbrace{H(p_i)}_{\substack{(i)\\ \text{Prediction confidence}}} + \underbrace{\lambda_{zs}H(p_i,\hat{p}_i)}_{\substack{(ii)\\ \text{Zero-shot regularization}}}\Big) - \underbrace{\lambda_\beta H(\beta)}_{\substack{(iii)\\ \text{Entropy regularization}}} \tag{4}$$

(i) Prediction confidence. This term minimizes the entropy of the prediction vectors p_i for each sample i, encouraging confident predictions across the N_S samples.

(ii) Zero-shot regularization. The purpose of this term is to prevent the prediction vectors p_i from deviating significantly from an initial prediction $\hat{p}_i$. Typically, we can obtain $\hat{p}_i$ with the single template *This is a sound of* $\{$*class*$\}$ which we name the zero-shot prediction.

(iii) Entropy regularization. This term is an entropy-based regularizer for the shared weight vector β, commonly used in the literature [14–17], which encourages smoother and non-sparse weight distributions. This term is outside the summation as β is common to all samples. In addition, the convex penalty $-\lambda_\beta H(\beta)$ serves as an entropy barrier that implicitly enforces $\beta_j \geq 0$.

2.4. Optimization procedure

The pseudo-code of the optimization procedure is outlined in Algorithm 1. It should be noted that all embeddings f and t are computed only once before the optimization. At initialization, the prompt template weights are initialized with a uniform distribution and the zero-shot prediction vectors $\hat{p}$ are computed. The weights β are updated according to an iterative fixed-point update rule derived from minimizing the objective function until convergence (i.e. solving $\frac{\partial\mathcal{L}}{\partial\beta_j} = 0$):

$$\beta_j^{(t)} = \frac{\exp(R_j^{(t)}/\lambda_\beta)}{\sum_{l=1}^{N_T}\exp(R_l^{(t)}/\lambda_\beta)} \tag{5}$$

with $R_j^{(t)}$ being interpreted as the sum of the contribution from each individual sample i:

$$R_j^{(t)} = \sum_{i=1}^{N_S}\Bigg[\sum_{k=1}^{N_C} p_{ik}\Big(\ln(p_{ik}) + H(p_i)\Big)l_{ijk} + \lambda_{zs}\sum_{k=1}^{N_C} p_{ik}\Big(\ln(\hat{p}_{ik}) - E_{p_i}[\ln(\hat{p}_i)]\Big)l_{ijk}\Bigg] \tag{6}$$

The contribution can be decomposed into two terms. The first term is linked to the entropy of p_i, while the second term evaluates the similarity with the zero-shot prediction, with λ_{zs} balancing the contribution of the two terms.

Algorithm 1: Optimization Procedure

Input: $\mathbf{f}, \mathbf{t}, \tau$
$\varepsilon \leftarrow 1 \times 10^{-6}$; ▷ Stopping threshold
Initialize $\beta^{(0)} = \frac{1}{N_T}\mathbf{1}_{N_T}$; ▷ Uniform weights
Compute $\hat{p}_i \quad \forall i$; ▷ Zero-shot predictions
while $\|\beta^{(t)} - \beta^{(t-1)}\|_2 \geq \varepsilon$ **do**
  $t = t + 1$;
  Compute $p_i \quad \forall i$; ▷ See Eq. (3)
  Compute $R_j^{(t)} \quad \forall j$; ▷ See Eq. (6)
  Update $\beta^{(t)}$; ▷ See Eq. (5)
return β ; ▷ Optimized weights

3. EXPERIMENTAL SETUP

3.1. Evaluation datasets

We evaluate our method on five benchmark audio classification datasets spanning three different tasks: environmental, urban, and vocal sound classification. For environmental classification, we use ESC-50 [18], which contains $2k$ audio clips of 5 seconds each across 50 classes, and ESC-Actions [18], a subset of the previous dataset with 400 clips covering 10 action-related classes. For urban sound classification, we use US8K [19], comprising $\sim$ $8k$ clips of $\sim$ 4 seconds each across 10 classes, and SESA [20], composed of $\sim$ 600 clips of $\sim$ 30 seconds each across 4 surveillance-related classes. For vocal sound classification, we employ VS [21], consisting of $\sim$ $21k$ clips grouped into 6 classes, with each clip being a 5-second recording. For all five datasets, we use the entire dataset as a single test set given that we are working in a zero-shot setting.

3.2. Text prompts

We construct the set of templates using $N_T = 35$ hand-crafted prompt templates commonly employed in the audio–language model literature [5–7, 12]. These templates comprise a variety of sentence structures, detailed in Table 1. The same set of templates is used across all five evaluation datasets. The relevance of this set of templates for prompt ensemble methods is highlighted by the observed high variability in accuracy across all five benchmark datasets (see Figure 2).

3.3. Implementation details

We use the original CLAP-2022 [5] audio–language model to generate the audio embeddings $\mathbf{f}$ and all text embeddings $\mathbf{t}$. We set the base model logit scale to 33.3, as per model specification. Concerning our method's hyperparameters, we fix λ_β to 0.01 throughout our experiments. As for λ_{zs}, we set it to 100 for Experiment 1 (single-sample β) and 0.1 for Experiment 2 (dataset β). This choice is further discussed in Section 4.2.

3.4. Baselines

We implement six different baseline prompt ensembling methods to compare our approach with:

(i) Majority voting consists of selecting the most frequently predicted class across the entire set of prompt templates for each sample. All prompt templates' votes are weighted equally.

Table 1: Prompt templates used in our experiments. Class name should replace {}.

Declarative (This is...): This is a sound of {}, This is an audio of {}, This is an audio clip of {}, This is a sound clip of {}, This is an audio track of {}, This is a sound track of {}, This is an example of {}, This is {}
Descriptive (A/An...): A sound of {}, An audio of {}, A recording of {}, A sound recording of {}, An audio recording of {}, A sound clip of {}, An audio clip of {}, An audio track of {}, A sound track of {}, A sound snippet of {}, An audio snippet of {}
Imperative (Listen/Hear...): Listen to {}, Listen to the sound of {}, Listen to an audio of {}, Listen to a recording of {}, Listen to a sound recording of {}, Listen to an audio recording of {}, Hear the sound of {}, Hear an audio of {}
Direct Noun: Sound of {}, Audio of {}, Recording of {}, Sound recording of {}, Audio recording of {}, Audio clip of {}
Miscellaneous: I can hear {}, {}

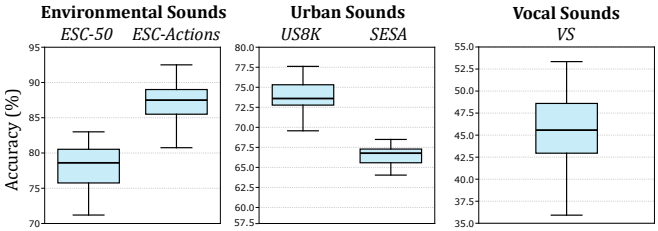

Fig. 2: Distribution of accuracies obtained from 35 hand-crafted prompt templates across five audio datasets. It reveals substantial variability to prompt selection for all datasets.

(ii) Majority voting with entropy-weighting, in contrast, weights each template's vote by the inverse of the entropy of its prediction vector. Thus, more confident template predictions have greater influence on the final classification.

(iii) Majority voting with pruning consists of removing 50% of the prompt templates with the highest entropy before performing a majority vote on the rest.

(iv) Average text embedding consists of calculating individual embeddings for each prompt template and then computing the normalized sum of these embeddings for each sample (i.e. β is uniformly distributed). The resulting average text embedding is used for classification.

(v) Average text embedding with entropy-weighting modifies this process by weighting each text embedding in the normalized sum by the inverse of the entropy of its corresponding prediction vector. Therefore, text embeddings derived from confident templates contribute more significantly to the final embedding used for classification.

(vi) Average text embedding with pruning consists of removing 50% of the prompt templates with the highest entropy before calculating individual text embeddings for remaining prompt template and then computing the normalized sum of these embeddings for each sample. The resulting average text embedding is used for classification.

Table 2: Classification accuracy of our proposed approach on five audio classification datasets. We report three different settings: single-sample β (i.e., $N_S = 1$), dataset β (i.e., N_S equals the size of the dataset), and dataset β with pruning (see Section 4.3 for more details) and compared against six baseline ensemble methods (see Section 3.4).

Method	Environmental Sounds		Urban Sounds		Vocal Sounds	Average
	ESC-50	*ESC-Actions*	*US8K*	*SESA*	*VS*	
Zero-Shot Prediction						
"This is a sound of {class}"	82.6	87.7	75.0	66.7	46.9	71.8
Baseline Ensemble Methods						
Majority voting	80.7	89.5	76.3	66.4	47.2	72.0
Majority voting with entropy-weighting	81.2	90.3	75.1	66.6	47.4	72.1
Majority voting with pruning	82.9	90.3	75.1	67.1	46.0	72.3
Average text embedding	80.5	90.0	76.0	66.6	47.6	72.1
Average text embedding with entropy-weighting	80.6	90.0	76.2	66.6	47.6	72.2
Average text embedding with pruning	82.7	90.3	75.1	67.1	46.6	72.3
Proposed Approach						
Single-sample β	82.9	90.0	74.9	**67.2**	47.7	72.5
Dataset β	83.5	90.5	77.3	66.8	47.9	73.2
Dataset β with pruning	**83.5**	**91.5**	**77.4**	66.8	**48.9**	**73.6**

4. EXPERIMENTS

4.1. Experiment 1: Single-sample β

In this experiment, we consider the scenario where we have access to one test sample at a time ($N_S = 1$) and, therefore, compute an individual weighting vector β per sample. The results are shown in Table 2. We observe encouraging improvements, with an average gain of up to 0.7% compared to the zero-shot setting, outperforming all baseline ensemble methods. In particular, on ESC-50 we outperform the average text embedding baseline by 2.4%. This highlights our capability to find an optimized β weighting vector with just a *single sample*, surpassing classical average weighting of text embeddings.

4.2. Experiment 2: Dataset β

In this experiment, we have access to the entire evaluation dataset and compute a common weighting vector, β, for all samples. We decrease the value of λ_{zs} to 0.1, reducing the need for zero-shot regularization, as we compute β with a larger number of samples. An ablation study on the λ_{zs} parameter is shown in Table 3 further motivating our choice. As seen in Table 2, we obtain average gains of 0.7% compared to the single-sample β optimization and average gains of 0.9% over all baseline methods.

4.3. Experiment 3: Dataset β with pruning

In this experiment, we perform multiple pruning cycles on the weighting vector β. During each cycle, we optimize β before pruning it. The pruned β vector is used to initialize the next cycle. We apply 4 cycles with a pruning percentage of 15% at each cycle ending up with an overall pruning percentage of $\sim 50\%$. Additionally, we conduct an ablation study, depicted in Table 4, to further validate our choice. As shown in Table 2, we achieve further average gains of 0.4% compared to dataset-level β optimization, resulting in cumulative average gains of 1.4% across all baseline ensemble methods. Notably, the best performances are observed in 4 out of 5 datasets, with ESC-Actions achieving gains of up to 3.8% over its zero-shot prediction. Note that even with multiple pruning cycles, the runtime of the proposed approach is comparable to the baseline methods, with a runtime of 0.2s (see Table 5), which remains negligible compared to the time required to encode both audio and text features.

Table 3: Average accuracy across all datasets for different values of λ_{zs}, highlighting the need for further regularization toward the zero-shot prediction in the single-sample optimization case. Choice highlighted in blue.

	λ_{zs}					
	0.01	*0.1*	*1*	*10*	*100*	*1000*
Single-sample β	72.18	72.22	72.38	72.39	**72.48**	72.22
Dataset β	72.79	**73.19**	71.33	70.84	70.62	70.61

Table 4: Average accuracy across all datasets for different pruning percentages per cycle and total cycles. Choice highlighted in blue.

Pruning% / Cycle	Total Cycles				
	1	*2*	*3*	*4*	*5*
5	73.26	72.92	73.13	73.33	73.30
10	72.96	73.31	73.54	73.42	73.45
15	73.22	73.35	73.52	**73.60**	73.56
20	73.37	73.45	73.43	73.34	73.20
25	73.46	73.42	73.38	72.21	72.12

Table 5: Runtime of our approach compared to feature extraction and baseline methods. Evaluation was performed with 35 templates on a 24 GB NVIDIA GeForce RTX 4090 GPU using the VS dataset ($\sim 21k$ audio samples split into six classes).

	Features encoding	Baselines	Proposed Approach
Runtime	$\sim$ 2 min	$\sim$ 0 s	$\sim$ 0.2 s

5. CONCLUSION

We propose an entropy-guided prompt weighting algorithm for zero-shot audio–language classification that strategically combines embeddings from multiple templates, using low-entropy predictions as a proxy for high confidence. We demonstrate that this approach consistently outperforms baseline ensemble methods across five benchmark audio classification datasets. The proposed method offers a practical solution that can be integrated into existing audio-language model zero-shot classification pipelines, alleviating the need for manual prompt engineering when encountering new datasets.

References

[1] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.

[2] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen, "Clotho: An audio captioning dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[3] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. ICML, 2021, pp. 8748–8763.

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

[6] Benjamin Elizalde, Soham Deshmukh, and Huaming Wang, "Natural language supervision for general-purpose audio representations," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 336–340.

[7] Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang, "Pengi: An audio language model for audio tasks," *Advances in Neural Information Processing Systems*, vol. 36, pp. 18090–18108, 2023.

[8] Ching-Feng Yeh, Po-Yao Huang, Vasu Sharma, Shang-Wen Li, and Gargi Gosh, "Flap: Fast language-audio pre-training," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[9] Xuenan Xu, Zhiling Zhang, Zelin Zhou, Pingyue Zhang, Zeyu Xie, Mengyue Wu, and Kenny Q Zhu, "Blat: Bootstrapping language-audio pre-training based on audioset tag-guided synthetic data," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 2756–2764.

[10] Asif Hanif, Maha Tufail Agro, Mohammad Areeb Qazi, and Hanan Aldarmaki, "Palm: Few-shot prompt learning for audio language models," *arXiv preprint arXiv:2409.19806*, 2024.

[11] Nishit Anand, Ashish Seth, Ramani Duraiswami, and Dinesh Manocha, "Tspe: Task-specific prompt ensemble for improved zero-shot audio classification," in *2025 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, 2025, pp. 1–5.

[12] Michel Olvera, Paraskevas Stamatiadis, and Slim Essid, "A sound description: Exploring prompt templates and class descriptions to enhance zero-shot audio classification," *arXiv preprint arXiv:2409.13676*, 2024.

[13] James Urquhart Allingham, Jie Ren, Michael W Dusenberry, Xiuye Gu, Yin Cui, Dustin Tran, Jeremiah Zhe Liu, and Balaji Lakshminarayanan, "A simple zero-shot prompt weighting technique to improve prompt ensembling in text-image models," in *International Conference on Machine Learning*. ICML, 2023, pp. 547–568.

[14] Ségolène Martin, Yunshi Huang, Fereshteh Shakeri, Jean-Christophe Pesquet, and Ismail Ben Ayed, "Transductive zero-shot and few-shot clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28816–28826.

[15] Maxime Zanella, Benoît Gérin, and Ismail Ayed, "Boosting vision-language models with transduction," *Advances in Neural Information Processing Systems*, vol. 37, pp. 62223–62256, 2024.

[16] Maxime Zanella, Fereshteh Shakeri, Yunshi Huang, Houda Bahig, and Ismail Ben Ayed, "Boosting vision-language models for histopathology classification: Predict all at once," in *International Workshop on Foundation Models for General Medical AI*. Springer, 2024, pp. 153–162.

[17] Karim El Khoury, Maxime Zanella, Benoît Gérin, Tiffanie Godelaine, Benoît Macq, Saïd Mahmoudi, Christophe De Vleeschouwer, and Ismail Ben Ayed, "Enhancing remote sensing vision-language models for zero-shot scene classification," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[18] Karol J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. pp. 1015–1018, ACM Press.

[19] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1041–1044.

[20] Tito Spadini, "Sound events for surveillance applications," oct 2019, October 2019.

[21] Yuan Gong, Jin Yu, and James Glass, "Vocalsound: A dataset for improving human vocal sounds recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 151–155.