



# Domain Adaptation for Contrastive Audio-Language Models

Soham Deshmukh<sup>1,2</sup>, Rita Singh<sup>1</sup>, Bhiksha Raj<sup>1</sup>

<sup>1</sup>Carnegie Mellon University, <sup>2</sup>Microsoft

{sdeshmuk, rsingh, bhikshar}@andrew.cmu.edu

## Abstract

Audio-Language Models (ALM) aim to be general-purpose audio models by providing zero-shot capabilities at test time. The zero-shot performance of ALM improves by using suitable text prompts for each domain. The text prompts are usually hand-crafted through an ad-hoc process and lead to a drop in ALM generalization and out-of-distribution performance. Existing approaches to improve domain performance, like few-shot learning or fine-tuning, require access to annotated data and iterations of training. Therefore, we propose a test-time domain adaptation method for ALMs that does not require access to annotations. Our method learns a domain vector by enforcing consistency across augmented views of the testing audio. We extensively evaluate our approach on 12 downstream tasks across domains. With just one example, our domain adaptation method leads to 3.2% (max 8.4%) average zero-shot performance improvement. After adaptation, the model still retains the generalization property of ALMs.

**Index Terms:** test-time adaptation, zero-shot audio classification, audio-language models

## 1. Introduction

Audio-Language Models (ALMs) use language as a mode of supervision to learn general-purpose audio representations [1, 2, 3, 4]. Once pretrained, the ALMs can be used for multiple downstream tasks. For instance, the widely used paradigm of CLAP [2, 5] contrastively trains an audio and text encoder on millions of audio-text pairs and at test time shows impressive generic zero-shot performance. Then the model is employed in a zero-shot setup across various domains from Sound Event Classification to Speech Emotion Recognition. For each domain, human input in crafting text prompts helps in improving the zero-shot performance of ALMs.

The human prompt engineering improves the ALM's performance on the specific domain. For example, manual prompt engineering [6] from "*{class}*" to "*this is a sound of {class}*" leads to a relative 2% improvement in performance on audio event datasets. The text prompts are designed for each domain using an ad-hoc experimental process. This process generally requires knowledge of the ALM's training data which may or may not be available to the public. Existing approaches improve domain performance by Few-Shot Learning (FSL). This involves learning prompts [7] or adapters [8] using supervised training on annotated target data. The annotation process is expensive and may not be feasible for deployed models. FSL methods also lead to a loss in generalization of base ALM and poor out-of-distribution performance [9, 8].

Parallely, in the vision domain, there have been two model adaptation frameworks: FSL and Test-Time Training [10]. The

examples for FSL are CoOp [11] and CoCoOp [11], both of which require access to labels. On the other hand, Test-Time Training [10] focuses on using test examples and auxiliary tasks for full or partial model [12, 13] and prompt updates [14]. Test-time tuning is a promising direction because it does not make assumptions about target data distribution and does not require labeled data. However, the existing Test-Time Training methods require either an additional self-supervision branch, multiple test-time examples, or suffer from high computation costs due to parameter updates. Specifically for audio, there has not been an exploration of test-time adaptation for ALMs, let alone compute and parameter-efficient test-time updates.

In this work, we propose a domain adaptation method for Contrastive ALMs that is performed at test-time using unlabeled audio. Our method involves learning a domain vector, which helps the model adjust to new audio domains by enforcing consistent prediction across augmented views of audio. This process is divided into three stages: Augment: The test audio is subject to various augmentations to simulate different listening conditions. Alongside this, a domain vector is incorporated into the model's text processing component; b) Combine: The model makes predictions based on the augmented audio and the modified text embeddings. These predictions are then averaged to produce a more stable and generalized output, and c) Optimize: Finally, the averaged prediction probabilities are used to compute self-entropy to fine-tune the domain vector, ensuring that the model's predictions are adapted for the domain. With only one unlabelled audio, we see an average 3.2% (max 8.4%) zero-shot performance improvement across 12 downstream tasks, while retaining the generalization property of ALMs.

## 2. Background

### 2.1. Contrastive Audio-Language Models

This section provides a generic overview of Contrastive Audio-Language models in literature. We will commonly refer to them as "model". The model has a two-tower structure consisting of audio and text branches. The two branches are pretrained using contrastive learning [15]. After pretraining, the model can be used for zero-shot classification. To illustrate, the user provides an audio file and the  $N$  classes to classify the audio into. The audio file is converted into a mel-spectrogram ( $x \in \mathbb{R}^{T \times F}$ ) and passed to an audio encoder to produce audio embeddings  $v \in \mathbb{R}^{1 \times d}$ . On the text side,  $N$  classes are tokenized and embeddings are obtained for each token. Then the token embeddings are passed through a text encoder to produce text embeddings  $u \in \mathbb{R}^{d \times N}$ . Both embeddings are projected into a common embedding space using two independent projection layers. After projection, the dot product is computed between the two embeddings. Generally, an activation function like Softmax or sigmoid

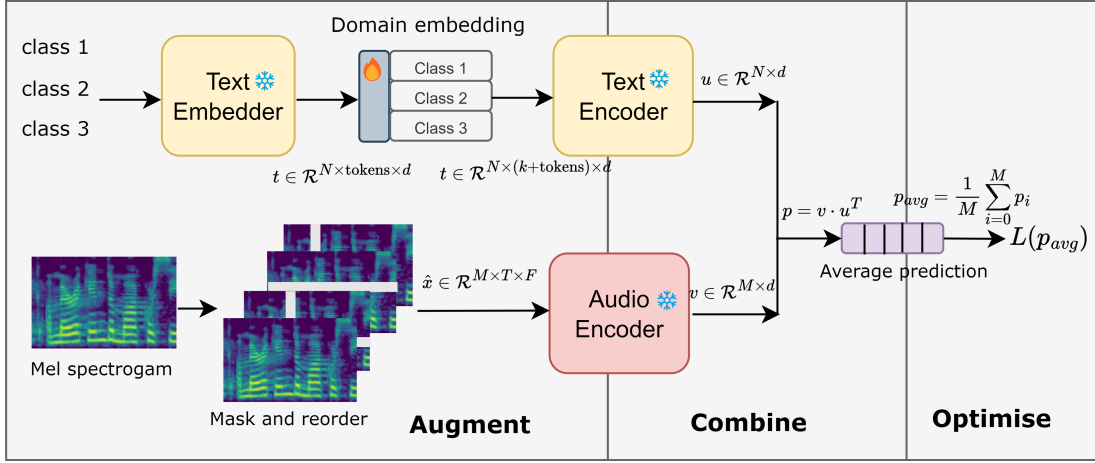


Figure 1: Our method takes a single input audio at test time and optimizes a domain embedding using an entropy-based loss function. The method does not require labels and enforces consistent prediction across augmented views.

is applied to obtain a valid probability distribution across  $N$  classes of interest.

## 2.2. Human-crafted text-prompts

The contrastive models like CLAP [6] append prompts to classes for better zero-shot performance. The prompts are hand-crafted for the domain the model is being tested on. For example, a prompt of "this emotion is " is used for Speech Emotion Recognition while the prompt of "this is the sound of " is used for Sound Event Classification. The choice of text prompt for each task affects performance [6, 16]. From the prompt experiment, we hypothesize that: *though audio-language models have an adequate understanding of audio concepts, we don't have methods to elicit this information from the models.* The manual method of hand-crafting prompts is cumbersome and requires domain knowledge. Moreover, the handcrafted prompts don't take into account intra-task variance resulting from domain changes. For example, zero-shot inference for the task of sound event classification can provide different performances when evaluated in an urban city, a domestic household versus in a rainforest. This leads us to believe an automatic unsupervised adaptation method for zero-shot inference of audio-language can help improve performance.

## 3. Methodology

Our approach can be summarized as learning a domain vector  $a$  that minimizes the entropy of prediction probability obtained from multiple audio augmentations for test audio:

$$\min_a -p \log p$$

where  $p$  is the average augmentation probability distribution:  $\frac{1}{M} \sum_{i=1}^M p(y_i | z_i)$ ,  $z_i$  is the  $i^{\text{th}}$  augmentation for test audio  $x$ ,  $y_i$ . The method is depicted in Figure 1 and is divided into three parts: Augment, Combine, and Optimize.

### 3.1. Augment

The audio file is converted to a mel spectrogram ( $x \in \mathbb{R}^{T \times F}$ ). Rather than directly passing  $x$  to the audio encoder, it is first augmented using multiple random yet controlled augmentations. The augmentation is masking-based and inspired by SpecAug [17]. The four main augmentations performed are:

**Time Masking (TM):** Random portions of spectrograms are masked along the time axis. The width of the mask ranges from

2 to 128. The number of masked stripes ranges from 2 to 24.

**Frequency Masking (FM):** Random portions of spectrograms are masked along the freq. axis. The width of the mask ranges from 2 to 32. The number of masked stripes ranges from 2 to 24.

**Time & Frequency Masking (TFM):** Random portions of spectrograms are masked along both the frequency and time axis. This is the combination of time masking followed by frequency masking.

**Time Reorder (TR):** Audio-language models compress all audio information into a single embedding which leads to loss of temporal information [18]. Therefore, we explicitly add a time-reorder augmentation (details in appendix).

All the augmentation along with the original input  $x$  is concatenated to form a batch of inputs:

$$\hat{x} = [\text{TM}(x), \text{FM}(x), \text{TFM}(x), \text{TR}(x), x]$$

where  $\hat{x} \in \mathbb{R}^{M \times T \times F}$  where  $M$  is number of augmentations plus one original spectrogram. The augmented and batch of spectrograms  $\hat{x}$  is passed through the audio encoder and linear projection to obtain audio embedding  $v \in \mathbb{R}^{M \times d}$ .

For the text side, the  $N$  classes are first tokenized and converted to token-level embeddings. Then a learnable domain vector is concatenated with the tokenized embeddings. The learnable domain vector is randomly initialized. If the token-level text embedding is  $t \in \mathbb{R}^{N \times \text{tokens} \times d}$  and is concatenated with the learnable vector of  $k$  tokens, the output embedding is  $t \in \mathbb{R}^{N \times (\text{tokens} + k) \times d}$ . Therefore, the method optimizes only one vector for the domain and the one vector does not vary per class. The  $t$  embedding is passed through a text encoder to produce text embeddings  $u \in \mathbb{R}^{d \times N}$ . For our discussion, we consider the independent projection layers as a part of audio and text encoder respectively.

### 3.2. Combine

The audio embedding  $v \in \mathbb{R}^{M \times d}$  and text embeddings  $u \in \mathbb{R}^{d \times N}$  are used to compute a dot product:

$$p = v \cdot u^T$$

where  $p \in \mathbb{R}^{M \times N}$ . We use softmax to convert  $p$  into a valid probability distribution across classes  $N$ . After softmax,  $p$  is the probability of each class for a total of  $M$  augmented views of the audio. The probability distribution  $p$  is averaged along the different augmentations  $M$  to get:  $p_{avg} = \frac{1}{M} \sum_{i=1}^M p_i$  where  $p_{avg}$  in  $\mathbb{R}^N$

	Average $\uparrow$	Sound Event Classification $\uparrow$			Vocal Sound Classification $\uparrow$	Surveillance Sound Classif. $\uparrow$	Acoustic Scene Classification $\uparrow$
Model	Average	ESC50	US8K	DCASE17 Task 4	Vocal Sound	SESA	TUT 2017
Zero-Shot	62.93	93.90	82.30	46.60	79.97	64.95	53.80
One audio	64.94	93.35	<b>85.26</b>	50.96	82.14	73.30	54.19
Five audio	<b>65.92</b>	<b>95.05</b>	85.21	<b>52.30</b>	<b>82.40</b>	<b>74.35</b>	<b>54.38</b>

	Music Classification $\uparrow$		Instrument Classification $\uparrow$		Speech Emotion Classification $\uparrow$	
Model	GTZAN Music	GTZAN Speech	Beijing Opera	NS Instr. family	CRE MA-D	RAV DESS
Zero-Shot	99.20	58.40	46.60	68.00	30.00	31.54
One audio	99.21	61.00	47.45	68.28	29.92	<b>34.25</b>
Five audio	<b>1.00</b>	<b>63.20</b>	<b>50.42</b>	<b>69.23</b>	<b>31.27</b>	33.19

Table 1: Zero-shot and Test-time adaptation performance on 12 downstream tasks. The adaptation method uses one randomly chosen unlabelled audio example and five randomly chosen unlabelled audio examples at test time. The Table reports the average numbers across 5 runs. The metric is accuracy and higher is better for all tasks. Average  $\uparrow$  is the average performance across 12 tasks

### 3.3. Optimise

The  $p_{avg}$  is the average of augmented probability distributions over  $N$  classes. To learn the domain vector, we have to use a loss function. As the labels are not known, the constructed loss function has to be a self-supervision or unsupervised loss. Therefore, we choose self-entropy as the loss function. By minimizing self-entropy, we enforce consistency across the predictions of augmented views. Therefore, the loss function is:

$$\mathcal{L} = -p_{avg} \log p_{avg}$$

The loss  $\mathcal{L}$  is calculated using a single test example and optimizes a parameter of size:  $k \times d$ , where  $k$  is the number of tokens.

## 4. Experiments

### 4.1. Downstream tasks

We benchmark our method on 12 downstream tasks [19] across the domains of Sound Event Classification, Acoustic Scene Classification, Vocal Sounds, Music, Surveillance, and Speech Emotion Recognition. The datasets used are: ESC50 [20], Urban-Sound8K [21], DCASE2017 Task4[22], TUT 2017, GTZAN Music Speech [23], GTZAN Genres [23], Beijing Opera Percussions [19], CREMA-D [24], RAVDESS [25], Vocal Sound [26]. The datasets have varied audio duration, classes, files, and setups. For example, the audio duration ranges from 3 to  $\geq 35$  seconds, classes range from binary to 50 classes, audio files per data range from 120 to 305k files.

### 4.2. Implementation details

**Contrastive ALM.** We use the SoTA Contrastive Audio-Language Models [1] for our experiments. The audio encoder is HTSAT[27] and the text encoder is a modified GPT2 [28, 29]. The audio is sampled at 44.1 kHz and converted to log Mel spectrograms with 64 Mel bins, a hop size of 320, and a window size of 1024 in the range of 50-14000 Hz range. We use the zero-shot setup of ALM as the baseline.

**Test-Time Adaptation.** We use a single token of dimension 768 as the learnable domain vector. Relating to Section 3.3, this implies  $k$  is 1 and  $d$  is 768. The optimizer used is Weighted Adam [30] with a learning rate  $5e-2$ .

## 5. Results

The Results are organized as follows: Section 5.1 covers the domain adaptation results, Section 5.2 describes the effect of

number of augmentations, Section 5.3 studies cross-domain generalization, and Section 5.4 discusses limitation of the proposed method.

### 5.1. Test-time domain adaptation

The main results are shown in Table 1. As this is the first test-time adaptation method for audio, the zero-shot performance of the model to be adapted (CLAP) is considered baseline performance. The adaptation technique using only one example, provides an average relative improvement of 3.18% in zero-shot performance. When 5 samples are used for adaptation, the improvement increases to 4.7%.

For some downstream tasks like SESA, the improvement is 8.35% and 9.4% for 1 and 5 example adaptation respectively. The SESA task consists of 16 kHz audio with up to 33 seconds with classes Casual, Gunshot, Explosion, and Siren (also contains alarms). However, the audio is hard to distinguish even for humans, for example, the audio marked Casual could be a fireworks, factory machine, or thunder which might perceptual sound like other threat classes. The adaptation method forces consistency among predictions for different augmented views, i.e. it learns a vector that makes the model robust to different views of audio in the wild. We hypothesize that this is helpful for scenarios where the sound is perceptually similar and requires more fine-grained prompts to elicit the right information from audio-language models for classification. The downstream task of DCASE2017Task4 has similar perceptually hard-to-distinguish sounds, where the adaptation method helps to improve zero-shot performance by 4.36% and 5.7%.

For the downstream task of Speech Emotion Recognition (SER), specifically dataset CREMAD, the adaptation method leads to a drop in zero-shot performance (-0.08%) when provided with one example. CLAP is not trained on speech data and performs poorly on SER in a zero-shot setup. For example, the zero-shot performance is 30%, the random performance is 17% and the supervised performance is 75.2%. The CLAP zero-shot performs poorly when the audio encoder has not learned relevant features for the tasks. In such cases, our method leads to minor or no improvements, such as in Speech Emotion Recognition and Keyword Word Spotting.

### 5.2. Effect of Augmentations

For the results in Table 1, the audio files are augmented 50 times using augmentation strategies described in 3.1. The larger the

Dataset	ZS	ESC50	US8K	D17T4	SESA	GT MS	Genre	Opera	CREM	RAVD	Vocal	TU17	NSyn	Avg.
ESC50	93.90	93.35	81.84	46.74	72.16	100.0	60.60	40.56	26.19	31.16	81.18	54.26	67.46	62.96
US8K	82.30	90.80	85.26	49.93	67.35	97.66	54.40	44.07	23.57	32.47	76.74	45.05	62.71	60.83
D17T4	46.60	90.60	84.72	50.96	69.40	96.88	54.70	44.92	26.86	26.96	76.02	45.48	62.81	60.86
SESA	64.95	92.40	81.45	45.26	73.30	99.22	56.30	33.47	28.02	32.22	81.87	51.60	65.26	61.70
GT MS	99.20	90.75	85.01	49.70	65.64	99.21	54.70	36.02	24.96	27.59	75.91	45.31	60.42	59.60
Genre	58.40	92.55	82.38	46.15	72.60	98.44	61.00	36.44	29.54	32.14	82.21	49.75	64.05	62.27
Opera	46.60	92.40	82.56	46.74	73.01	99.22	56.59	47.45	27.61	32.30	82.46	53.58	64.50	63.20
CREM	30.00	92.60	81.92	46.07	73.58	99.22	56.69	36.02	29.92	32.38	82.18	52.59	64.99	62.35
RAVD	31.54	92.50	81.37	45.41	74.46	99.22	59.10	37.71	29.39	34.25	82.04	52.16	65.23	62.74
Vocal	79.97	92.75	81.88	46.07	74.09	99.22	58.19	36.44	28.29	32.22	82.23	51.91	64.97	62.36
TU17	53.80	92.40	81.39	45.33	73.92	99.22	56.39	33.47	28.02	32.26	81.87	54.19	65.21	61.97
NSyn	68.00	92.50	81.16	45.56	73.58	99.22	55.75	34.32	27.32	32.22	81.51	51.36	68.28	61.90
Avg.	62.93	92.13	82.58	46.99	71.92	90.64	57.03	38.41	27.47	31.51	80.52	50.61	64.66	61.21

Table 2: Cross-domain generalization. The rows indicate the dataset from which one sample is used to learn the domain vector. The columns indicate the target dataset used for evaluation. The ZS column is the baseline zero-shot performance of CLAP. The blue diagonal indicates the performance when the sampled audio used to learn the domain vector, comes from the target domain.

number of augmentation, the more views the adaptation method can utilize to improve performance. To study the effect of augmentation, we perform an ablation study where only an audio file undergoes half the augmentation leading to a smaller number of augmented views. The comparison results are shown in Table 3. On average, the larger number of augmentation helps when either one example or five examples are used for adaptation. Therefore, even with 25 augmented views, the domain adaptation improves over a zero-shot baseline. The higher number of augmentations (views) helps in improving performance on harder tasks. Therefore, with fewer augmentation, we see a drop in performance on SESA where audio is perceptually hard to classify. The number of augmentations used is a hyperparameter and can be tuned and determined based on the compute power available at inference.

	One test example	Five test example
25 Augmentations	63.51	63.77
50 Augmentations	<b>64.94</b>	<b>65.92</b>

Table 3: Effect of number of augmentations on domain adaptation performance for one example and five example setup.

### 5.3. Cross-domain generalization

The base ALM is a general purpose zero-shot model. The adaptation of ALM to a specific domain, should not lead to a drastic drop in its zero-shot performance. To verify if this holds for our method, we setup an ablation study. We adapt the base ALM CLAP on a target domain and check performance on other domains. For example, we adapt CLAP to ESC50 (dataset 1 in Table 1) and check performance on remaining datasets (dataset {2 - 12} in Table 1). This process can be repeated  $T$  times, where  $T$  is the number of target domains, to provide an estimate of drop in zero-shot performance. The numerical results of the experiments are shown in Table 2. The diagonal entries (highlighted in blue), correspond to adaptation performed using one unlabelled audio sampled from the target domain data.

As we are interested in drop in generalization of ALM compared to the gain in the target domain performance, we summarize the findings from Table 2 in Fig 2. On average, we observe a 3% increase in domain performance for an average zero-shot performance drop of 1%. The exception is domain adaptation performed on ESC50, which slightly improves the average zero-shot performance.

### 5.4. Limitations

Our adaptation method improves zero-shot performance by using one unlabelled audio at test-time. Its practical applications

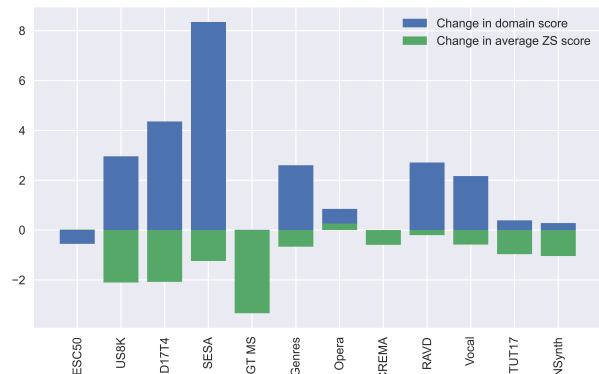


Figure 2: Cross-domain adaptation performance. The x-axis indicates the dataset used for domain adaptation. The upper and lower bar indicate the change in - domain performance and average zero-shot score respectively.

include on-device updates for audio-language model model to adapt it to the user’s audio and media. In this setup, access to labeled annotations is not possible but the model does have access to test audio used for inference. However, compared to zero-shot inference, the method incurs a higher computational cost because of backpropagation used to update the domain vector and the processing of multiple augmented views. Depending on the downstream application and the on-device compute, the parameter update may not be feasible, limiting the applicability of this method.

## 6. Conclusion

We propose a domain adaptation method for ALMs. The method uses unlabelled audio at test-time to enforce consistent prediction across augmented views of audio by optimizing a domain vector. It consists of three parts: Augment, Combine, and Optimise. Augment: The test audio is subject to various augmentations to simulate different listening conditions. Alongside this, a domain vector is incorporated into the model’s text processing component; b) Combine: The model makes predictions based on the augmented audio and the modified text embeddings. These predictions are then averaged to produce a more stable and generalized output, and c) Optimize: Finally, the averaged prediction probabilities are used to compute self-entropy to fine-tune the domain vector, ensuring that the model’s predictions are adapted for the domain. Adaption with one unlabelled audio provides an average 3.2% (max 8.4%) zero-shot improvement and with 5 audios an average improvement of 4.7% (max 9.4%). After adaptation, the ALM still retains its zero-shot generalization.

## 7. References

- [1] B. Elizalde, S. Deshmukh, and H. Wang, "Natural language supervision for general-purpose audio representations," *arXiv preprint arXiv:2309.05767*, 2023.
- [2] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," *arXiv preprint arXiv:2211.06687*, 2022.
- [3] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An audio language model for audio tasks," *arXiv preprint arXiv:2305.11834*, 2023.
- [4] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. Glass, "Listen, think, and understand," *arXiv preprint arXiv:2305.10790*, 2023.
- [5] 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," *arXiv preprint arXiv:2303.17395*, 2023.
- [6] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. 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.
- [7] Y. Li, X. Wang, and H. Liu, "Audio-free prompt tuning for language-audio models," *arXiv preprint arXiv:2309.08357*, 2023.
- [8] J. Liang, X. Liu, H. Liu, H. Phan, E. Benetos, M. D. Plumbley, and W. Wang, "Adapting language-audio models as few-shot audio learners," *arXiv preprint arXiv:2305.17719*, 2023.
- [9] H.-H. Wu, O. Nieto, J. P. Bello, and J. Salamon, "Audio-text models do not yet leverage natural language," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [10] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *International conference on machine learning*. PMLR, 2020, pp. 9229–9248.
- [11] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.
- [12] Y. Gandelman, Y. Sun, X. Chen, and A. Efros, "Test-time training with masked autoencoders," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 29 374–29 385. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/bcdec1c2d60f94a93b6e36f937aa0530-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/bcdec1c2d60f94a93b6e36f937aa0530-Paper-Conference.pdf)
- [13] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=uX13bZLkr3c>
- [14] M. Shu, W. Nie, D.-A. Huang, Z. Yu, T. Goldstein, A. Anandkumar, and C. Xiao, "Test-time prompt tuning for zero-shot generalization in vision-language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 274–14 289, 2022.
- [15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021.
- [16] S. Deshmukh, D. Alharthi, B. Elizalde, H. Gamper, M. A. Ismail, R. Singh, B. Raj, and H. Wang, "Pam: Prompting audio-language models for audio quality assessment," *arXiv preprint arXiv:2402.00282*, 2024.
- [17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, 2019.
- [18] H.-H. Wu, O. Nieto, J. P. Bello, and J. Salomon, "Audio-text models do not yet leverage natural language," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [19] J. Turian, J. Shier *et al.*, "HEAR: Holistic Evaluation of Audio Representations," in *NeurIPS 2021 Competitions and Demonstrations Track*, 2022.
- [20] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018.
- [21] J. Salamon, C. Jacoby *et al.*, "A dataset and taxonomy for urban sound research," in *22nd ACM international conference on Multimedia*, 2014.
- [22] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.
- [23] G. Tzanetakis, G. Essl, and P. Cook, "Automatic musical genre classification of audio signals," 2001. [Online]. Available: <http://ismir2001.ismir.net/pdf/tzanetakis.pdf>
- [24] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-D: Crowd-sourced emotional multimodal actors dataset," *IEEE transactions on affective computing*, vol. 5, no. 4, pp. 377–390, 2014.
- [25] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PLOS ONE*, vol. 13, no. 5, pp. 1–35, 05 2018. [Online]. Available: <https://doi.org/10.1371/journal.pone.0196391>
- [26] Y. Gong, J. Yu, and J. Glass, "Vocalsound: A dataset for improving human vocal sounds recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [27] 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 *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [28] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [29] S. Deshmukh, B. Elizalde, D. Emmanouilidou, B. Raj, R. Singh, and H. Wang, "Training audio captioning models without audio," *arXiv preprint arXiv:2309.07372*, 2023.
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [31] R. Zhou, S. Deshmukh, J. Greer, and C. Lee, "Narle: Natural language models using reinforcement learning with emotion feedback," *arXiv preprint arXiv:2110.02148*, 2021.
- [32] H. Dharmyal, B. Elizalde, S. Deshmukh, H. Wang, B. Raj, and R. Singh, "Describing emotions with acoustic property prompts for speech emotion recognition," *arXiv preprint arXiv:2211.07737*, 2022.
- [33] Y. Chu, J. Xu, X. Zhou, Q. Yang, S. Zhang, Z. Yan, C. Zhou, and J. Zhou, "Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv:2311.07919*, 2023.
- [34] J. Wang, Z. Du, Q. Chen, Y. Chu, Z. Gao, Z. Li, K. Hu, X. Zhou, J. Xu, Z. Ma, W. Wang, S. Zheng, C. Zhou, Z. Yan, and S. Zhang, "Lauragpt: Listen, attend, understand, and regenerate audio with gpt," *arXiv preprint arXiv:2310.04673*, 2023.
- [35] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 716–23 736, 2022.
- [36] C. Kim, J. Park, H. Shim, and E. Yang, "SGEM: Test-time adaptation for automatic speech recognition via sequential-level generalized entropy minimization," in *Conference of the International Speech Communication Association (INTERSPEECH)*, 2023.
- [37] Y. Wang, N. J. Bryan, M. Cartwright, J. Pablo Bello, and J. Salamon, "Few-shot continual learning for audio classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 321–325.