PALM: Few-Shot Prompt Learning for Audio Language Models

Asif Hanif, Maha Tufail Agro, Mohammad Areeb Qazi, Hanan Aldarmaki
Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI)
{asif.hanif, maha.tufail, mohammad.qazi, hanan.aldarmaki}@mbzuai.ac.ae

Abstract

Audio-Language Models (ALMs) have recently achieved success in zero-shot audio recognition tasks, which match features of audio waveforms with class-specific text prompt features, inspired by advancements in Vision-Language Models (VLMs). Given the sensitivity of zero-shot performance to the choice of hand-crafted text prompts, many prompt learning techniques have been developed for VLMs. We explore the efficacy of these approaches in ALMs and propose a novel method, *Prompt Learning in Audio Language Models (PALM)*, which optimizes the feature space of the ALM text encoder. Unlike existing methods that work in the input space, our approach results in greater training efficiency. We demonstrate the effectiveness of our approach on 11 audio recognition datasets, encompassing a variety of speech-processing tasks, and compare the results with three baselines in a few-shot learning setup. Our results show that PALM performs on a par with or outperforms the baselines while being more computationally efficient. Our code is publicly available at Github[†].

1 Introduction

Inspired by the success of Vision-Language Models (VLMs) (Zhang et al., 2024), Audio-Language Models (ALMs) have recently emerged, achieving state-of-the-art performance on various zero-shot audio recognition tasks (Elizalde et al., 2023; Deshmukh et al., 2023; Kong et al., 2024; Das et al., 2024). In zero-shot audio recognition, features of the audio waveform are matched with features of text prompts representing each class, and the highest matching class is assigned to the audio waveform. Zero-shot audio recognition offers some advantages by eliminating the need for extensive labeled datasets and allowing for the recognition of new classes without additional training. This approach reduces training times and data annotation costs, leading to substantial savings in computational resources.

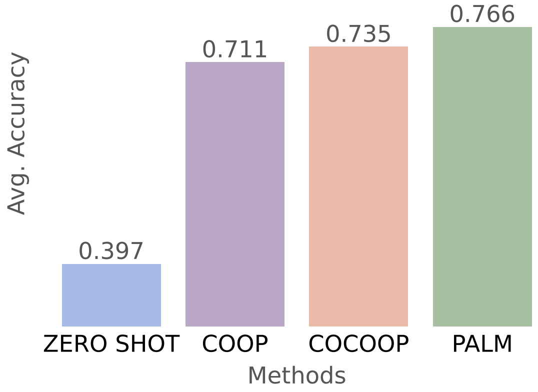


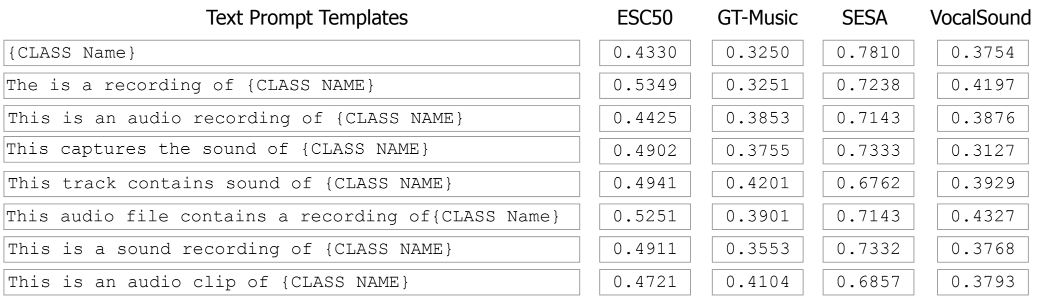
Figure 1: Comparison of our proposed approach, PALM, with three baselines: ZERO-SHOT (Deshmukh et al., 2023), COOP (Zhou et al., 2022b) and COCOOP (Zhou et al., 2022a). Bar plots show classification accuracy averaged across 11 audio datasets encompassing various speech-processing tasks.

The choice of text prompt is crucial for pre-trained vision-language and audio-language models, but it becomes a drawback for zero-shot recognition due to the requirement of hand-crafted prompts. This manual prompt-engineering can result in performance variations (Zhou et al., 2022b,a). We confirm this observation, previously noted in VLMs, within the context of ALMs (refer to Figure 2). To automate the learning of text prompts, various approaches have been introduced for prompt learning in VLMs (Gu et al., 2023).

The domain of prompt learning in ALMs remains under-explored, lacking comprehensive studies to evaluate the effectiveness of prompt learning techniques within this context. To bridge

[†] https://asif-hanif.github.io/palm/

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 18527–18536
November 12-16, 2024 ©2024 Association for Computational Linguistics



Text Prompt Templates	ESC50	GT-Music	SESA	VocalSound
{CLASS Name}	0.4330	0.3250	0.7810	0.3754
The is a recording of {CLASS NAME}	0.5349	0.3251	0.7238	0.4197
This is an audio recording of {CLASS NAME}	0.4425	0.3853	0.7143	0.3876
This captures the sound of {CLASS NAME}	0.4902	0.3755	0.7333	0.3127
This track contains sound of {CLASS NAME}	0.4941	0.4201	0.6762	0.3929
This audio file contains a recording of{CLASS Name}	0.5251	0.3901	0.7143	0.4327
This is a sound recording of {CLASS NAME}	0.4911	0.3553	0.7332	0.3768
This is an audio clip of {CLASS NAME}	0.4721	0.4104	0.6857	0.3793

Figure 2: **Impact of Hand-crafted Prompts on ZERO-SHOT Performance** Zero-shot accuracy across four audio recognition datasets (ESC50 (Piczak), GT-Music-Genre (Sturm, 2012), SESA (Spadini, 2019), and VocalSound (Gong et al., 2021)) is evaluated with eight different text prompts using PENGI (Deshmukh et al., 2023) model. The accuracy varies with changes in the handcrafted prompts.

this research gap, we adapt prompt learning techniques developed for VLMs and apply them to the domain of ALMs. Our results demonstrate that these adaptations improve audio classification performance (see Table 2). Traditional techniques optimize the input space (i.e. token embeddings) of the text encoder by introducing a learnable context. However, this approach can increase training costs as loss gradients must flow through the text encoder branch. To address this, we introduce a novel method, **PALM**: **P**rompt Learning in **A**udio **L**anguage **M**odels, which optimizes the feature space of the text encoder rather than its input space. This makes the training computationally efficient since loss gradients do not need to flow through the text encoder. To assess the effectiveness of our approach, we show results on 11 audio recognition datasets, encompassing various speech processing tasks. Our method either matches or surpasses other approaches while being less computationally demanding (see Table 2 and Table 3).

Contributions: Our contributions are as follows:

- Inspired by the success of few-shot prompt learning in vision-language models (VLMs), we show that adapting these techniques to audio-language models (ALMs) significantly enhances their performance.

- We introduce PALM, a novel few-shot prompt learning method for ALMs that optimizes the text encoder's feature space. We demonstrate our approach's effectiveness on 11 audio recognition datasets, comparing it to three baselines in a few-shot learning setup. Our method matches or outperforms others while being less computationally demanding, establishing a benchmark for prompt learning in ALMs and paving the way for future research.

2 Related Work

Prompt engineering involves adding task-specific hints, called prompts, to a large pre-trained model to adapt it to new tasks. Recently, significant advancements have been made in prompt learning, particularly in the fields of language and vision. Below, we outline the recent developments in language, vision, and audio domains.

2.1 Audio Language Models (ALMs)

Taking inspiration from multimodal models like CLIP (Radford et al., 2021) in the vision domain, Contrastive Language-Audio Pretraining (CLAP) (Elizalde et al., 2023) stands out as the first-of-its-kind audio language model. It connects natural language and audio through dual encoders and contrastive learning, aligning audio and text descriptions in a shared multimodal space. Furthermore, CLAP introduces zero-shot prediction capabilities, removing the necessity for training with predefined class labels and allowing flexible class prediction during inference.

PENGI (Deshmukh et al., 2023), another audio language Model, utilizes transfer learning by treating all audio tasks as text-generation tasks. It takes audio recordings and text inputs, generating free-form text as output. The input audio is represented by continuous embeddings from an audio encoder, while the corresponding text input undergoes the same process with a text encoder. These sequences are combined as a prefix to prompt a pre-trained frozen language model. PENGI's unified architecture supports both open-ended and close-ended tasks without requiring additional fine-tuning or task-specific extensions.

Audio Flamingo, introduced by Kong et al. (2024), is a multimodal-to-text generative model inspired by Flamingo (Alayrac et al., 2022), demonstrating advanced audio understanding capabilities, adaptability to unseen tasks through in-context learning and retrieval, and multi-turn dialogue abilities. The model features an audio feature extractor with a sliding window and uses cross-attention to fuse audio inputs into the language model, ensuring computational efficiency.

2.2 Prompt Learning in Language Models

Extensive research has been conducted on prompt learning techniques in natural language processing. Pioneering work by (Brown et al., 2020) focused on optimization strategies for zero-shot and few-shot learning scenarios, demonstrating that prompts can enable generative models to perform well across various tasks without extensive task-specific training. Their method leverages the model's pre-trained knowledge and prompt-guided interactions to achieve strong performance on new tasks. They also introduced GPT-3, which transformed the field of prompt learning in natural language processing. Petroni et al. (2019) integrated contextual cues and constraints within prompts to guide model behavior, embedding task-specific information to enhance output precision and relevance. Their technique improves interpretability and task-oriented performance by providing contextual guidance during inference.

2.3 Prompt Learning in Vision-Language Models

Inspired by advancements in prompt-based work in language models, several studies have been conducted to adapt these methods to VLMs (Gu et al., 2023). Some focus exclusively on the language component, such as COOP (Context Optimization) (Zhou et al., 2022a). In contrast, others integrate insights from language and visual components, as seen in COCOOP (Conditional Context Optimization) (Zhou et al., 2022a). COOP enhances CLIP model's few-shot transfer learning capability by optimizing a continuous set of prompt vectors within the language branch. However, COCOOP addresses the limitations of COOP, particularly its suboptimal performance on novel classes, by explicitly conditioning prompts on individual image instances, thereby enhancing generalization.

2.4 Prompt Learning in Audio-Language Models

Prompt learning with audio-language models is relatively understudied. Previous work has explored enhancing language models with speech recognition by conditioning them on variable-length audio embeddings using a conformer-based audio encoder (Fathullah et al., 2024). Deshmukh et al. (2024) propose a test-time domain adaptation method for Contrastive ALMs, using unlabeled audio to adjust the model to new domains via a domain vector, consistent predictions, and self-entropy fine-tuning, improving on traditional Test-Time Training. Li et al. (2024) introduce *PT-Text*, an audio-free prompt tuning scheme that optimizes prompt tokens from text, regularizing the model to avoid overfitting by training with captions and using a multi-grained strategy to enhance performance.

3 Method

3.1 Audio-Language Model (ALM)

We demonstrate the efficiency of prompt learning in enhancing zero-shot performance using a state-of-the-art audio-language model PENGI (Deshmukh et al., 2023). Our approach is applicable to all audio-language models that have *aligned* audio and text encoders.

PENGI takes an audio recording/waveform and a text prompt as input and generates free-form text as output. It consists of three branches. The first branch is an audio encoder that maps the audio waveform to an embedding space. The second branch is a text encoder that transforms the input text into the same embedding space. These embeddings are then concatenated to form an input prefix for the third branch, a causal language model that generates tokens autoregressively, conditioned on both the audio and text inputs. PENGI can be used for various audio-conditioned tasks, such as text completion, classification, audio caption generation, and question-answering (Deshmukh et al., 2023).

Zero-Shot Inference Although PENGI is multimodal-to-text generation model, however, we use its audio and text encoder branches for zero-shot audio recognition. This is accomplished by comparing the embedding of the audio waveform (extracted from the audio encoder) with the embeddings of text prompts for different classes (extracted from the text encoder). An overview of zero-shot inference is given in Figure 3(a). It should be noted that the zero-shot setup used by (Deshmukh et al., 2023) differs from ours, as they employ the model's free-form text output for zero-shot inference.

Formally, we denote the pre-trained ALM as $f_\theta = \{f_A, f_T\}$, whereas f_A and f_T are audio and text encoders, respectively and θ represents the combined weights of both encoders. For classification in zero-shot scenario, an audio waveform $\mathbf{x}$ is first passed to the audio encoder f_A to produce a $d-$dimensional feature vector $f_A(\mathbf{x}) \in \mathbb{R}^d$. In parallel, text prompts representing each class label $y_i \in \{y_1, y_2 \ldots, y_c\}$ are encapsulated within class-specific handcrafted text templates, such as

$$t_i = \text{“}\texttt{An audio recording of } \{\text{CLASS } y_i\}\text{”},$$

where c is the total number of classes. Each text prompt, represented as t_i, is processed through the text encoder f_T, resulting in a feature vector $f_T(t_i) \in \mathbb{R}^d$. The relationship between the audio waveform $\mathbf{x}$ and a class-specific text prompt t_i is quantified by computing the cosine similarity between their corresponding feature vectors, denoted as $\texttt{sim}(f_A(\mathbf{x})\ ,\ f_T(t_i))$. The class with the highest similarity score is then assigned as the label $\hat{y}$ for the audio waveform, i.e.

$$\hat{y} = \underset{i \in \{1,2,\ldots,c\}}{\mathbf{argmax}} \ \ \texttt{sim}\big(f_A(\mathbf{x})\ ,\ f_T(t_i)\big). \qquad (1)$$

3.2 PALM: Prompt Learning in ALM

In our proposed method, we do not use handcrafted prompts; instead, we simply use class names as the input to the text encoder i.e. $t_i =$ “$\{\text{CLASS } y_i\}$”. Moreover, unlike COOP (Zhou et al., 2022b), which learns the context of input text prompts in the token embedding space (see Figure 3(b)), we learn the context in the feature space of prompts. Specifically, after obtaining the feature vector of the i_{th} class text prompt via the text encoder, i.e., $f_T(t_i) \in \mathbb{R}^d$, we add a learnable vector $z_i \in \mathbb{R}^d$ to it to get the updated text feature vector as follows:

$$f'_T(t_i) = (1 - \lambda_i) \cdot f_T(t_i)\ +\ \lambda_i \cdot z_i \qquad (2)$$

where $\lambda_i \in [0, 1]$ is a learnable parameter that determines the contributions of both vectors. Assuming $\mathbf{t} = \{t_1, t_2, \ldots, t_c\}$ denotes text prompts of all classes, the raw/un-normalized prediction scores (logits), denoted as $f_\theta(\mathbf{x}, \mathbf{t}) \in \mathbb{R}^c$, for an audio waveform $(\mathbf{x})$ are obtained as follows:

$$f_\theta(\mathbf{x}, \mathbf{t}) = \Big\{\ \texttt{sim}\Big(f_A(\mathbf{x})\ ,\ f'_T(t_i)\Big)\ \Big\}_{i=1}^{c},$$

where $\texttt{sim}(\cdot)$ is cosine-similarity function and c is the number of classes. $f_A(\mathbf{x})$ is the feature vector from the audio encoder, and $f'_T(t_i)$ is the updated text feature vector (Equation 2) of i_{th} class.

We optimize the following objective function to learn feature-space context embeddings $\mathbf{z} = \{z_1, z_2, \ldots, z_c\}$ and their corresponding contributions $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_c\}$,

$$\underset{\mathbf{z}\ ,\ \lambda}{\mathbf{minimize}} \sum_{(\mathbf{x},y)\in\mathcal{D}} \mathcal{L}\big(f_\theta(\mathbf{x}, \mathbf{t}), y\big), \qquad (3)$$

where $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ is training dataset consisting of N audio-class pairs and $\mathcal{L}(\cdot)$ denotes cross-entropy loss. We use *few-shot* setting during training, meaning that a fixed number[†] of samples (e.g., 16) are randomly selected from each class in the training dataset. While optimizing objective in Equation 3, weights of both encoders $\{f_A, f_T\}$ are kept in frozen state. The number of learnable parameters in our proposed method is $c + (c \times d)$. After learning the parameters, we use Equation 4 for audio classification during inference stage.

$$\hat{y} = \underset{i \in \{1,2,\ldots,c\}}{\mathbf{argmax}} \ \ \texttt{sim}\big(f_A(\mathbf{x})\ ,\ f'_T(t_i)\big) \qquad (4)$$

An overview of our proposed approach can be found in Figure 3(c).

3.3 Difference with COOP and COCOOP

COOP (Zhou et al., 2022b) and COCOOP (Zhou et al., 2022a) were originally introduced for vision-language model; however, we adapted them for audio-language model (replacing the vision encoder branch with audio encoder branch) and presented it as baseline methods. Both of these baselines and our method aim to enhance zero-shot performance for audio classification in this work. While PALM and the baselines share this common goal, they differ in their approach to achieving it. COOP and COCOOP optimize the input space (token embeddings of prompt context) of text encoder, whereas PALM optimizes the text feature space.

[†] Refer to Figure 5 to see impact of number-of-shots on performance.

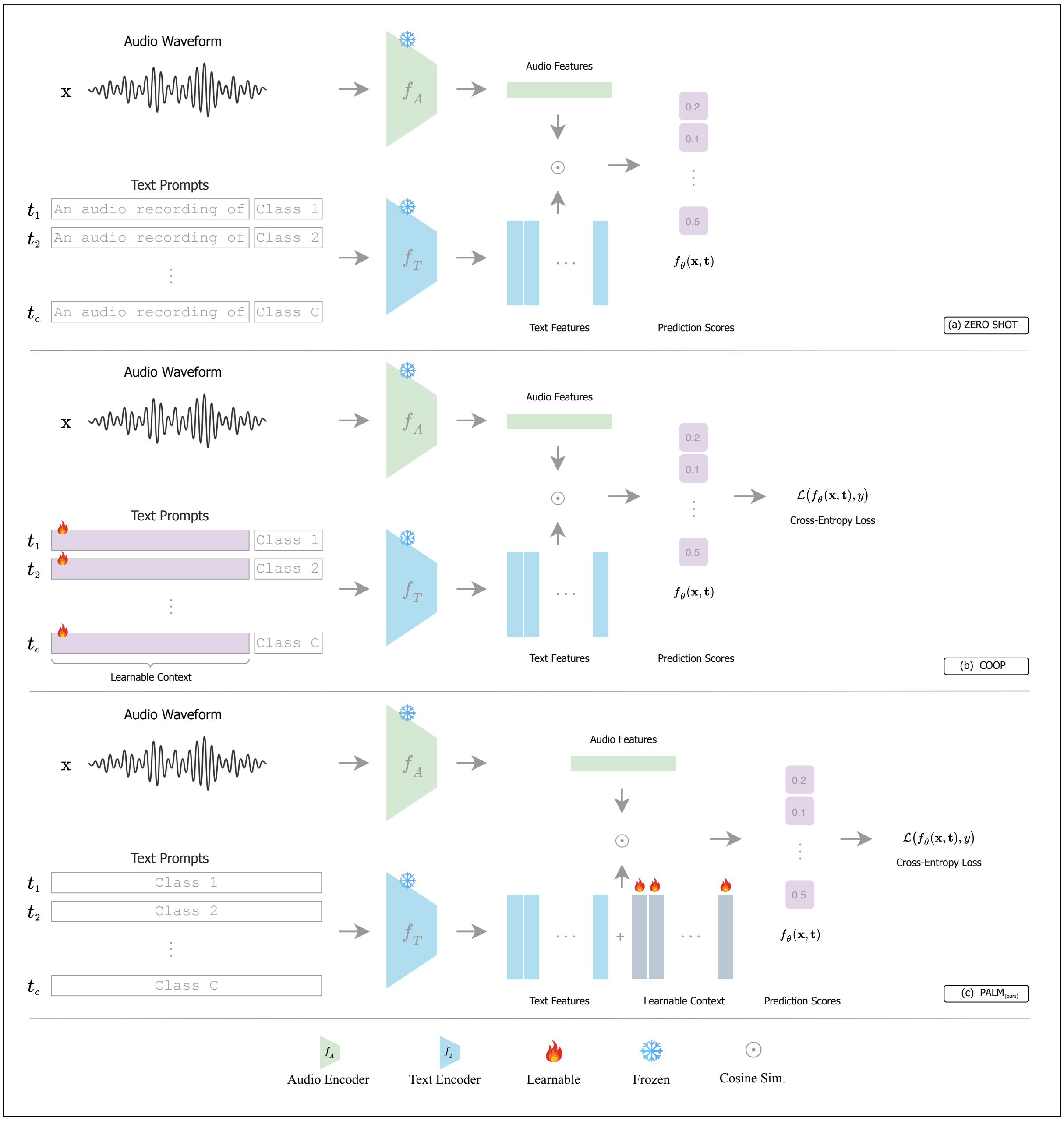


Figure 3: **Overview of Zero-Shot, COOP, PALM** *(a)* **Zero-Shot** inference involves matching the embedding of the audio waveform with the embeddings of text prompts for each class. The class with the highest matching score is then assigned to the audio. *(b)* **COOP** (Zhou et al., 2022b) avoids using handcrafted text prompts by learning the context of text prompts in the token embedding space. It optimizes the input space of the text encoder to enhance classification performance. *(c)* **PALM** requires only class names at the input of text encoder and it optimizes the feature space by adding learnable context embeddings to text feature vectors. PALM not only outperforms COOP (see Table 2), but it is also more computationally efficient since it does not require gradients to flow through the text encoder, unlike COOP.

In our method, loss gradients do not need to flow through the text encoder, whereas in COOP and COCOOP, gradients flow through the encoder to reach the input to update prompt context. Moreover, there is a feedback loop from audio features (output of audio encoder) to the input of text encoder in COCOOP, making it even more computationally expensive. Comparatively, PALM is more computationally efficient as it does not include a feedback loop (see Table 3). Both COOP and COCOOP require a user-specified hyper-parameter, namely the number of context tokens, whereas PALM does not rely on such a parameter. Results in Table 2 demonstrate that our method outperforms COOP

DATASETS	TYPE	CLASSES	SPLIT
Beijing-Opera	Instrument Classification	4	Five Fold
NS-Instruments		10	Train-Test
ESC50	Sound Event Classification	50	Five Fold
ESC50-Actions		10	Five Fold
UrbanSound8K		10	Ten Fold
CREMA-D	Emotion Recognition	6	Train-Test
RAVDESS		8	Train-Test
VocalSound	Vocal Sound Classification	6	Train-Test
SESA	Surveillance Sound Classification	4	Train-Test
TUT2017	Acoustic Scene Classification	15	Four Fold
GT-Music-Genre	Music Analysis	10	Train-Test

Table 1: **Datasets Information** In this work, we use 11 multi-class classification datasets encompassing a variety of speech-processing tasks.

and COCOOP, achieving an average improvement of 5.5% and 3.1% respectively.

4 Experiments and Results

4.1 Datasets

We evaluate our methodology using datasets from various speech-processing tasks: instrument classification, sound event classification, emotion recognition, vocal sound classification, surveillance sound event classification, acoustic scene classification, and music analysis. Brief information of each dataset can be found in Table 1. For instrument classification, we use Beijing-Opera (Tian et al., 2014) dataset, which includes audio examples of strokes from four percussion instrument classes used in Beijing Opera, and NS-Instruments (Engel et al., 2017) dataset, which consists of one-shot instrumental notes with unique pitches, timbres, and envelopes, spanning ten classes. For sound event classification, we utilize three datasets: ESC50 (Piczak), containing environmental recordings across 50 classes; ESC50-Actions (Piczak), a subset with 10 classes of non-speech human sounds; and UrbanSound8K (Salamon et al., 2014), with urban noise excerpts from 10 classes. Emotion recognition is assessed with the CREMA-D (Cao et al., 2014) and RAVDESS (Livingstone and Russo, 2018) datasets, covering 6 and 8 emotion classes respectively, performed by actors. We employ the VocalSound (Gong et al., 2021) dataset for vocal sound classification, which includes 6 classes of human non-speech vocalizations. For surveillance sound event classification, we use SESA (Spadini, 2019) dataset, which has 4 classes. Acoustic scene classification uses the TUT2017 (Heittola et al., 2017) dataset, containing samples from 15 acoustic scenes. For music analysis, the GT-Music-Genre (Sturm, 2012) dataset is used, which includes 10 classes of music genres.

We adhere to the official train-test or multi-fold splits for all datasets. We conduct cross-validation experiments on datasets having multi-fold splits such as Beijing-Opera, ESC50, ESC50-Actions, UrbanSound8K, and TUT2017, and report the average scores. We have publicly released all information regarding dataset preprocessing to ensure reproducibility of results.

4.2 Baseline Methods

For baselines, we consider PENGI model (Deshmukh et al., 2023) (in ZERO-SHOT setup), COOP (Zhou et al., 2022b) and COCOOP (Zhou et al., 2022a). COOP and COCOOP are prompt learning approaches, originally introduced for VLMs. Both of these approaches remove the requirement of providing handcrafted text prompts and they optimize the input token embedding space of text encoder to enhance accuracy. The only difference between COOP and COCOOP is that the latter incorporates a feedback loop from the output of the audio encoder to the input of the text encoder. We adapt these two approaches for audio-language models by replacing the vision encoder with an audio encoder and present them as baselines for our proposed method. *Why PENGI, COOP and COCOOP as baselines?* PENGI is an state-of-the-art ALM that has demonstrated comprehensive evaluation across 21 downstream audio tasks, making it a robust benchmark for comparison. COOP and COCOOP, on the other hand, are pioneering works on prompt learning in the domain of vision-language models, offering foundational techniques and insights that are highly relevant for our study.

4.3 Experimental Setup

We use pre-trained PENGI (Deshmukh et al., 2023) as the audio-language model for all methods. For all methods, except ZERO-SHOT, we conduct experiments for 50 epochs. Following the few-shot evaluation setup, we use 16 randomly selected samples per class from the training dataset. For inference, we utilize the entire test dataset. In the case of multi-fold datasets, we employ cross-validation and report the average scores. Training is performed using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.05. We use 'Accuracy' as the evaluation metric. For

METHODS →	ZERO SHOT	COOP				COCOOP				PALM$_{\text{(ours)}}$			
DATASETS ↓	-	SEED-0	SEED-1	SEED-2	AVG	SEED-0	SEED-1	SEED-2	AVG	SEED-0	SEED-1	SEED-2	AVG
Beijing-Opera	0.2881	0.9323	0.9660	0.9619	0.9534	0.9577	0.9830	0.9916	**0.9774**	0.9747	0.9066	0.9787	0.9533
CREMA-D	0.2310	0.3130	0.4197	0.2760	0.3362	0.2539	0.3358	0.3156	0.3018	0.4453	0.3580	0.2344	**0.3459**
ESC50-Actions	0.6525	0.9625	0.9400	0.9550	0.9525	0.9631	0.9620	0.9648	0.9634	0.9700	0.9625	0.9650	**0.9658**
ESC50	0.4965	0.9410	0.9390	0.9345	0.9382	0.9460	0.9370	0.9450	0.9427	0.9560	0.9600	0.9620	**0.9593**
GT-Music-Genre	0.3250	0.7250	0.6950	0.7350	0.7183	0.7500	0.7450	0.7600	0.7517	0.7900	0.7850	0.8250	**0.8000**
NS-Instruments	0.3291	0.5728	0.5562	0.6177	0.5822	0.5996	0.5740	0.6438	0.6058	0.6394	0.6108	0.6648	**0.6383**
RAVDESS	0.1222	0.3849	0.2688	0.3422	0.3320	0.3727	0.4399	0.3523	0.3883	0.4562	0.4603	0.4623	**0.4596**
SESA	0.7238	0.9143	0.8952	0.8762	0.8952	0.8381	0.8762	0.8952	0.8698	0.8857	0.9143	0.8857	**0.8952**
TUT2017	0.2435	0.6391	0.6667	0.6525	0.6528	0.7499	0.7215	0.7312	0.7342	0.7959	0.8047	0.7729	**0.7912**
UrbanSound8K	0.5349	0.7600	0.7378	0.7666	0.7548	0.7576	0.7784	0.7597	0.7652	0.8120	0.8037	0.8074	**0.8077**
VocalSound	0.4197	0.7162	0.7485	0.6642	0.7096	0.8081	0.7825	0.7463	0.7790	0.8101	0.8168	0.7964	**0.8078**
AVERAGE	0.3969	0.7146	0.7121	0.7074	0.7114	0.7276	0.7396	0.7369	0.7347	0.7759	0.7621	0.7595	**0.7658**

Table 2: **Comparison of** PALM **with Baselines** The accuracy scores of the methods (ZERO SHOT (Deshmukh et al., 2023), COOP (Zhou et al., 2022b), COCOOP (Zhou et al., 2022a), and our proposed PALM) across 11 audio recognition datasets are presented, with experiments (excluding ZERO SHOT) run using three seeds. Scores for each seed and the average are reported, with bold indicating the best average score in each row. Compared to the baselines, our proposed method achieves favorable results, with an average improvement of 5.5% over COOP and 3.1% over COCOOP.

METHOD	ZERO SHOT	COOP	COCOOP	PALM
# of Parameters	0	8,192	98,880	12,393

Table 3: **Number of Learnable Parameters** in baselines and PALM.

METHOD	AVERAGE ACCURACY
PALM + COOP	0.7236
PALM + COCOOP	0.7094
PALM + COCOOP†	0.7352
LINEAR PROBING	0.7299
PALM†	0.7160
PALM	**0.7658**

Table 4: **PALM+Baselines** Jointly optimizing input and output space of text encoder does not help attain better accuracy. COCOOP† refers to the method where feedback from audio features is incorporated into the text features, rather than being fed directly into the text encoder's input. PALM† denotes experiment in which text features are not used.

all methods, except ZERO-SHOT, we run experiments with three different seeds and report the scores for each seed along with the average score. For ZERO-SHOT, we use default text prompt template "`This is a recording of {CLASS NAME}`". For COOP (Zhou et al., 2022b) and COCOOP (Zhou et al., 2022a) baselines, we set the number context tokens to 16 and context is placed at the front of class names. PENGI model weights are kept "frozen" in all experiments. We use `NVIDIA A100-SXM4-40GB` GPU for all experiments and Pytorch version `1.11+cuda11.3`.

4.4 Results

Table 2 presents the performance comparison across 11 datasets using four different methods. Results indicate that PALM generally outperforms COOP and COCOOP, showing an average improvement of 5.5% over COOP and 3.1% over COCOOP. This suggests that PALM is a more effective approach in most cases. Moreover, it is important to note that PALM uses significantly fewer parameters—87% fewer compared to COCOOP. This reduction in parameters can contribute to more efficient model training and deployment.

In the datasets, namely Beijing-Opera, ESC50 and ESC50-Actions, the improvement of PALM over COCOOP is marginal. However, for the subsequent datasets, such as CREMA-D, GT-Music-Genre, NS-Instruments, RAVDESS, SESA, TUT2017, UrbanSound8K and VocalSound , the performance improvements are more substantial. This indicates that while PALM provides consistent benefits, the degree of performance gain varies across datasets.

4.5 Ablative Analysis

For ablative analysis, we first show the importance of incorporating learnable context embeddings in text features. In Figure 4, we compare the performance of our method with and without the learnable context embeddings. The results clearly demonstrate that removing the learnable context embeddings leads to a significant drop in performance, underscoring their crucial role in enhancing the model's accuracy. This highlights the effectiveness of our approach in optimizing the feature space of the text encoder.

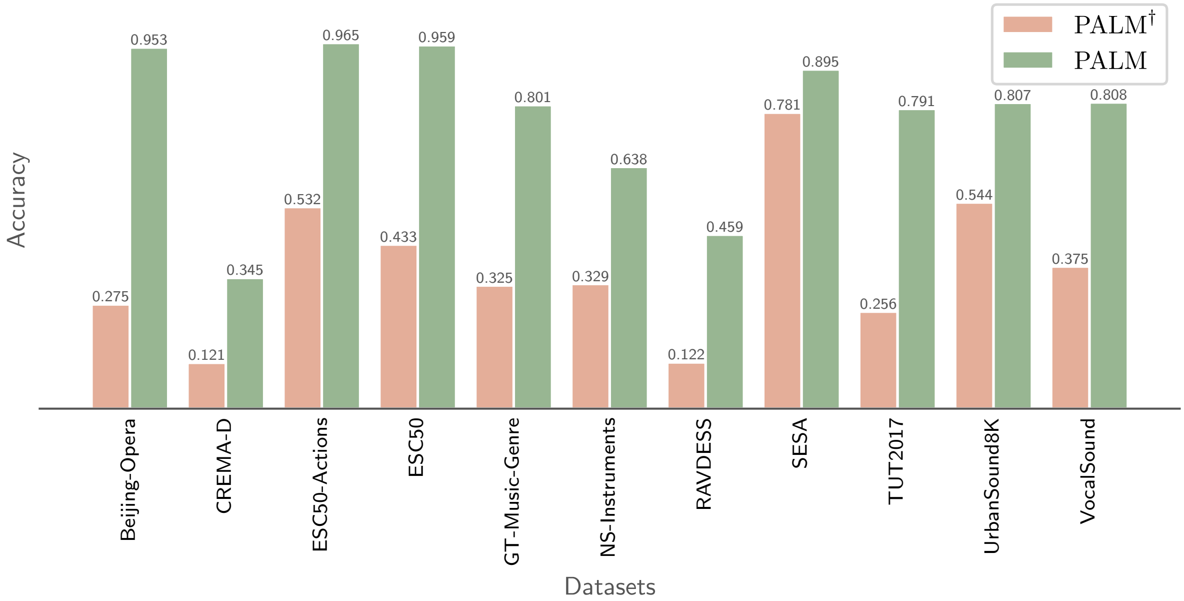


Figure 4: **Comparison of** PALM† **and** PALM. Here, PALM† refers to setting in which the *Learnable Context* embeddings (see Figure 3 for reference) have been **removed** from the feature space of the text encoder. The removal of context embeddings drastically degrades performance, highlighting their importance.

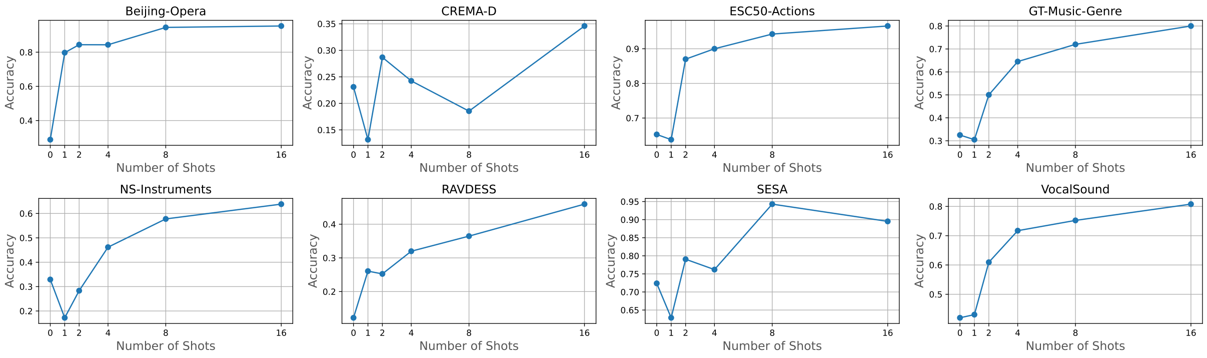


Figure 5: **Impact of *number-of-shots* on PALM's performance** A higher number of shots generally leads to increased audio classification accuracy using PALM.

We also demonstrate the effect of jointly optimizing the input and output spaces of the text encoder by applying PALM on top of COOP and COCOOP methods, as shown in the first two rows of Table 4. The results suggest that this joint optimization leads to a slight performance drop and is not beneficial. In the third row, we illustrate the impact of incorporating audio features into the text features, showing performance nearly identical to COCOOP, which integrates audio features into the text encoder's input space. Additionally, the fourth row presents the *linear-probing* (training a linear classifier on top of audio features) results, which underperform compared to PALM. Given that our approach is based on a few-shot setup, we also analyze the impact of the number of shots (training samples per class) on PALM's performance across eight datasets, as shown in Figure 5. As the number of shots in the training dataset increases, the performance of the model tends to improve. However, it should be noted that there is a trade-off between performance and computational load, as more shots also increase the computational requirements.

5 Conclusion

In this study, we investigate the application of prompt learning techniques, originally developed for vision-language models (VLMs), in the context of audio-language models (ALMs). We introduce PALM, a novel method that optimizes the feature space of the text encoder branch, enhancing training efficiency compared to existing methods that operate in the input space. Evaluated on 11 diverse audio recognition datasets, PALM consistently matches or surpasses established baselines in a few-shot learning setup while reducing computational demands. PALM offers a promising direction for enhancing the performance of ALMs in zero-shot and few-shot learning scenarios, contributing to the broader field of audio recognition and paving the way for future research in multimodal tasks.

Limitations

Although we are the first, to the best of our knowledge, to integrate prompt learning techniques originally designed for Vision-Language Models (VLMs) into Audio-Language Models (ALMs) and propose a new method, several aspects still need to be addressed. One critical aspect is to analyze prompt learning performance for domain generalization. This involves evaluating how well the prompts adapt to new, unseen domains and tasks, ensuring robustness and effectiveness across various applications. The second aspect is to analyze prompt learning performance under different types of perturbations in audio data to check its resilience against various types of noise. This analysis is essential for understanding the robustness of the models in real-world scenarios where audio data can be contaminated with background noise, distortions, and other audio artifacts. Thirdly, while our study shows results on audio classification, it is yet to be seen how prompt learning helps in other audio tasks such as speech recognition, audio segmentation, and information retrieval. Investigating the effectiveness of prompt learning across a broader range of audio tasks will provide a more comprehensive understanding of its potential and limitations. Fourthly, our few-shot method is specifically designed for single-label audio classification. Exploring a few-shot setup for multi-label audio classification and adapting PALM for this scenario remains an open question.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. 2014. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390.

Nilaksh Das, Saket Dingliwal, Srikanth Ronanki, Rohit Paturi, David Huang, Prashant Mathur, Jie Yuan, Dhanush Bekal, Xing Niu, Sai Muralidhar Jayanthi, et al. 2024. Speechverse: A large-scale generalizable audio language model. *arXiv preprint arXiv:2405.08295*.

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

Soham Deshmukh, Rita Singh, and Bhiksha Raj. 2024. Domain adaptation for contrastive audio-language models. *arXiv preprint arXiv:2402.09585*.

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

Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. 2017. Neural audio synthesis of musical notes with wavenet autoencoders.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, et al. 2024. Prompting large language models with speech recognition abilities. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13351–13355. IEEE.

Yuan Gong, Yu-An Chung, and James Glass. 2021. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip H. S. Torr. 2023. A systematic survey of prompt engineering on vision-language foundation models. *ArXiv*, abs/2307.12980.

Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen. 2017. TUT Acoustic Scenes 2017, Development dataset. Technical report, Department of Signal Processing, Tampere University of Technology.

Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. 2024. Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities. *arXiv preprint arXiv:2402.01831*.

Yiming Li, Xiangdong Wang, and Hong Liu. 2024. Audio-free prompt tuning for language-audio models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 491–495. IEEE.

Steven R Livingstone and Frank A Russo. 2018. 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*, 13(5):e0196391.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

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

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

Tito Spadini. 2019. Sound events for surveillance applications.

Bob L Sturm. 2012. An analysis of the gtzan music genre dataset. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 7–12.

Mi Tian, Ajay Srinivasamurthy, Mark Sandler, and Xavier Serra. 2014. A study of instrument-wise onset detection in beijing opera percussion ensembles. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 2159–2163. IEEE.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*.