Cacophony: An Improved Contrastive Audio-Text Model

Ge Zhu, *Graduate Student Member, IEEE*, Jordan Darefsky, and Zhiyao Duan, *Member, IEEE*

***Abstract*—Despite recent advancements, audio-text models still lag behind their image-text counterparts in scale and performance. In this paper, we propose to improve both the data scale and the training procedure of audio-text contrastive models. Specifically, we craft a large-scale audio-text dataset containing 13,000 hours of text-labeled audio, using pretrained language models to process noisy text descriptions and automatic captioning to obtain text descriptions for unlabeled audio samples. We first train on audio-only data with a masked autoencoder (MAE) objective, which allows us to benefit from the scalability of unlabeled audio datasets. We then train a contrastive model with an auxiliary captioning objective with the audio encoder initialized from the MAE model. Our final model, which we name Cacophony, achieves state-of-the-art performance on audio-text retrieval tasks, and exhibits competitive results on the HEAR benchmark and other downstream tasks such as zero-shot classification.**

***Index Terms*—Contrastive learning, joint audio-language embedding, self-supervised learning.**

I. Introduction

MACHINE audition [1] involves developing algorithms and systems for machines to analyze and understand sound, covering tasks such as audio tagging, acoustic scene classification, music classification, and sound event detection. In recent years, there has been a general shift away from individual audio pattern recognition tasks and toward general-purpose audio representations pretrained on large-scale audio datasets. Pretrained Audio Neural Networks (PANNs) [2] have played a significant role in this shift by demonstrating their versatility across various tasks and outperforming many advanced systems via fine-tuning.

Modern approaches aim for robust performance in general-purpose audio understanding tasks without the need for task-specific fine-tuning, which offers more flexibility. These methods approach general audio understanding by linking text and audio modalities, referred to as *audio-text models*. One approach to linking text and audio is through generating response text given a combination of an audio prompt and a text prompt [3], [4], [5]. For instance, Pengi [5] converts audio classification, retrieval, captioning, and audio question answering into a text generation task using audio and task-specific text prompts. Similarly, Qwen-Audio [6] addresses a variety of audio tasks through text generation but distinguishes itself by using text prompts consisting of hierarchical tag sequences inspired by Whisper [7]. Both Pengi and Qwen-Audio support multiple closed-ended and open-ended audio tasks without the need for additional fine-tuning or task-specific extensions of the architecture. Another method of linking text and audio is contrastive learning. Pretrained contrastive models can also be applied directly to various downstream tasks without fine-tuning. For instance, contrastive models can be used for retrieval and classification by assigning a score that identifies the most probable text (or class label) from a predefined set of choices for a given audio input. Moreover, the learned audio-text representations from contrastive models offer the flexibility to use one modality during training and the other at inference, which can be applied in text-to-audio generation [8], [9] and language-guided source separation [10], [11].

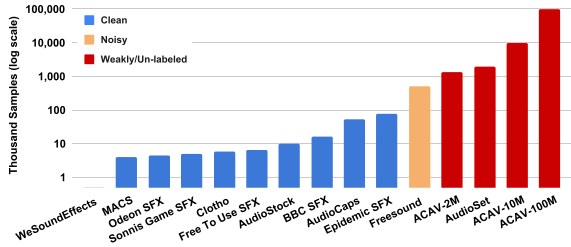

Fig. 1. A bar graph of the number of samples from commonly-used public audio datasets.

In this paper, we focus on improving contrastive audio-text models for general sounds by addressing two critical limitations in existing research: insufficient dataset scale and the vanilla contrastive training techniques. To tackle the dataset scale issue, we first examine the publicly available audio datasets. These can be categorized into three types based on label granularity: clean-labeled, noisy-labeled, and weakly-labeled or unlabeled, as illustrated in Fig. 1. Clean-labeled datasets, while high-quality, are limited in size. Noisy-labeled datasets offer more samples but include extraneous details. The largest category by far is weakly-labeled or unlabeled data, which provides little to no textual information. This scarcity of high-quality labeled

Received 30 May 2024; revised 29 September 2024; accepted 30 September 2024. Date of publication 23 October 2024; date of current version 21 November 2024. This work was supported in part by the New York State Center of Excellence in Data Science and in part by the University of Rochester Goergen Institute for Data Science seed funding Program. The associate editor coordinating the review of this article and approving it for publication was Zafar Rafii. *(Corresponding authors: Ge Zhu; Zhiyao Duan.)*

Ge Zhu was with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA. He is now with Adobe Research, San Jose, CA 95110-2704 USA (e-mail: gzhu@adobe.com, ge.zhu@rochester.edu).

Zhiyao Duan is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: zhiyao.duan@rochester.edu).

Jordan Darefsky is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA.

Digital Object Identifier 10.1109/TASLP.2024.3485170

data presents a significant challenge in audio-text contrastive learning, especially when compared to recent advancements in image-text models. For context, while the largest public audio-text datasets contain less than 100,000 pairs [12], [13], image-text models like CLIP [14] and SigLip [15] utilize 400 million and 3.6 billion pairs respectively.

Previous works have attempted to address this data scarcity issue by collecting data from various sources and applying natural language processing techniques to clean or filter noisy captions. For instance, Huang et al. [16] collected approximately 44 million 30-second music clips, applying a pretrained classifier and rule-based filtering to clean associated metadata. This process intensively reduced the dataset to 2 million music-text pairs. Notably, they found that models trained on large scale unfiltered audio-text data performed comparably to those trained on filtered data in music tagging and retrieval tasks, suggesting that data quantity might be as crucial as quality in this domain. Wu et al. [17], in collaboration with Large-scale Artificial Intelligence Open Network (LAION), curate the LAION-Audio dataset with 630K audio-text pairs together with 2 million Audioset [18] clips recaptioned with keyword-to-caption (K2C) augmentation to train LAION-CLAP. The K2C method uses a pretrained language model to generate captions from tags. However, the captions produced via K2C are restricted to the objects defined by the tags, offering limited descriptive details. K2C also risks making incorrect assumptions or introducing biases highlighted in [17]. Mei et al. [19] propose a multi-stage data filtering pipeline and utilize ChatGPT for cleaning text descriptions. Their contrastive language-audio pretraining (CLAP) models, trained on their WavCaps dataset, demonstrate superior performance in audio-text retrieval tasks compared to LAION-CLAP, despite a smaller-scale dataset. Such a text filtering pipeline reduces the amount of audio data, which could potentially result in reduced generalization.

In addition to dataset scale, there is a need for novel neural architectures and training strategies tailored to model audio structures more effectively. For instance, LAION-CLAP [17] investigates different choices of audio/text encoders, demonstrating superior performance with the hierarchical token semantic audio transformer (HTSAT) [20] for audio encoding and the Robustly optimized BERT[1] approach (RoBERTa) [21] for text encoding. In addition, LAION-CLAP proposes feature fusion for audio inputs with variable-length. In concurrent work, fast language-audio pretraining (FLAP) [22], inspired by fast language-image pretraining (FLIP) [23], proposes masking and removing a significant portion of spectrogram patches. FLAP also incorporates a reconstruction loss during the contrastive training on these masked spectrogram patches, although the improvements are modest.

In this paper, we investigate several strategies to improve the audio-text models, informed by the aforementioned challenges. For dataset creation, we collect a large-scale audio-text dataset and expand and refine its text descriptions. For audio recordings with weak or no labels, we utilize an automatic audio captioning model to obtain synthetic captions. For audio paired with noisy descriptions, we use large language models (LLMs) to generate several cleaned captions for each audio clip. These efforts lead to a collection of over 3.9 million audio-text pairs, with over 13,000 hours of audio.

For our neural architecture and training strategy, we propose to use a two-stage approach. The first stage focuses on training spectrogram-based audio encoder using a masked autoencoder (MAE) objective [24], [25], which learns representations through masking random patches from the input spectrogram and then reconstructing these masked patches. We anticipate that the MAE training will provide a better initialization for the following contrastive training. An audio classification objective, in contrast, may encourage the model to discard information unnecessary for classification but important for contrastive training. In the second stage, we use the audio encoder from the first stage to train our audio-text model on collected synthetic audio-text pairs, employing dual contrastive and captioning objectives. The integration of the auxillary captioning objective, inspired by contrastive captioner (CoCa) [26] and bootstrapping language-image pre-training (BLIP) [27], provides stronger supervision, encouraging the audio encoder to capture fine-grained patterns that closely match text descriptions. Training a captioner decoder also facilitates text generation for open-ended audio understanding tasks, expanding our model's application scope.

In the evaluation phase, incorporating a diverse range of evaluation tasks is crucial to comprehensively measuring model capability and preventing overfitting to common test sets, as suggested by Recht et al. [28]. Typically, audio-text representation learning is assessed through zero-shot audio classification and audio-text retrieval. To provide a more comprehensive benchmark, we additionally evaluate on audio question answering (AQA) [29]. To evaluate the effectiveness of our audio encoder, we test on Holistic Evaluation of Audio Representations (HEAR) [30]. HEAR offers broad evaluation tasks that test the general-purpose audio representation through audio classification and sound event detection. Lastly, to assess the performance of our captioning decoder, we evaluate on automatic captioning tasks for open-ended generation.

In summary, the contribution of our work includes: (1) We curate a large-scale refined audio-text dataset with LLM processing and audio captioning. (2) We propose a two-stage training approach for contrastive models: we first train an audio encoder with an MAE objective. In the second stage, we train a constrative model, initializing the audio encoder from the first stage, and include an auxiliary captioning objective to enhance the model's understanding of audio-text relationships. (3) We benchmark our model on a variety of audio understanding tasks. Particularly, our model achieves state-of-the-art or comparable performance on audio-text retrieval tasks. We have also conducted comprehensive ablation studies to demonstrate the impact of our different contributions. We open source the inference and evaluation codebase along with our pretrained model.[2]

[1]acronym for 'Bidirectional Encoder Representations from Transformers'

[2][Online]. Available: https://github.com/gzhu06/Cacophony

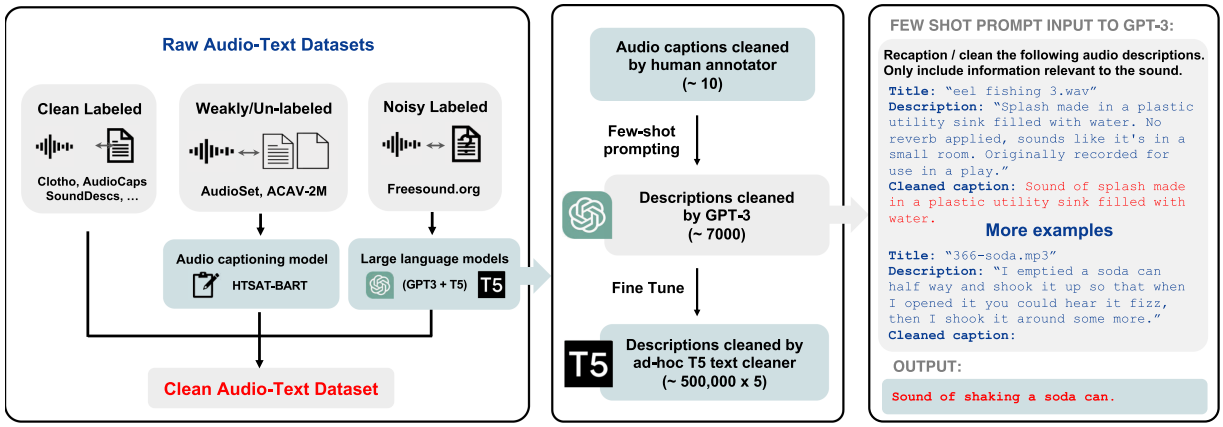

Fig. 2. Overview of our proposed dataset creation pipeline. Left: We process text descriptions based on the dataset quality. It aims to automatically clean and generate audio captions while maintaining consistency between the audio content and the textual descriptions. Middle: For datasets with raw, noisy descriptions, we use large language models, GPT-3 and T5-XXL, to clean the information that is irrelevant to the sound. Right: The detailed process of description cleaning using GPT-3 based on few-shot prompting is outlined as follows: Freesound raw inputs are highlighted in blue, sample inputs provided by a human annotator are marked in red, and the output text generated by GPT-3 is shown in bold red.

II. Dataset Collection

To make use of the audio data with either noisy or missing labels, we leverage publicly available tools to synthesize audio descriptions, shown in the left of Fig. 2. In the following, we propose strategies to effectively handle each case.

A. Clean-Labeled Datasets

The "clean-labeled" dataset split, which we refer to as "OpenSFX", comprises AudioCaps-train, Clotho-development, Epidemic Sound, SoundDescs, Free To Use Sounds, Sonniss Game Effects, AudioStock, and MACS. The most widely used datasets in this category are AudioCaps [12] and Clotho [13]. AudioCaps contains approximately 50,000 audio clips sourced from AudioSet [18] and annotated by humans. Clotho includes around 6,000 audio clips from Freesound,[3] with each clip featuring five human-generated captions. In addition to the commonly referenced datasets, there are others like the SoundDescs dataset [31], WavText5K [32], and Epidemic Sound.[4] The "writing styles" of the captions differ across audio-text datasets. Nevertheless, their raw descriptions all offer detailed content information about the audio clips. In our case, aiming to incorporate as much data as possible, we opt not to implement text-cleaning steps for processing these captions.

B. Noisy-Labeled Datasets

We also include a larger-scale audio data source, Freesound, which contains noisy text descriptions. Freesound is a collaborative online platform dedicated to sharing sounds, hosting over 500k audio clips uploaded by users, and has been used in studies such as WavCaps [19] and LAION-CLAP [17]. These clips, varying in duration, cover a wide range of audio content including music, environmental sounds, synthesized effects, and various noises. We exclude clips that are under one second or exceed five minutes, as shorter clips typically offer limited meaningful content and require excessive padding for training, and longer clips are often redundant. While Freesound prompts data uploaders give a brief description for each audio clip, these descriptions are often inaccurate and sometimes include named entities such as people's names, locations, and details about recording equipment.

We propose to use LLMs to transform raw Freesound descriptions into usable captions by automatically removing sound-irrelevant information. Although ChatGPT is well-suited for this task, the cost of generating captions for each Freesound sample would be prohibitively expensive. Given our resources, it would be more feasible to use a T5 model [33], but we find that available pretrained T5 models struggle with this task. We thus use GPT-3 to construct a small dataset of raw-clean pairs,[5] which will then be used to fine-tune T5-XXL [33] specifically for caption cleaning. This fine-tuned T5 model is subsequently used to transform the raw Freesound descriptions to clean captions for the entire Freesound dataset, as illustrated in the middle section of Fig. 2.

More specifically, to generate the synthetic dataset used for fine-tuning T5, we few-shot prompt GPT-3 to generate a clean caption given a noisy one with 10 human annotated pairs of noisy/clean captions in our prompt. This is shown in the right section of Fig. 2. We take a random sample of 7,000 (out of the total 500,000) Freesound noisy text captions and use GPT-3 to generate corresponding clean captions.

After fine-tuning the T5-XXL model with these pairs, we cleaned the text description for each audio sample in Freesound and generate five clean version; this servers as a form of data augmentation.

Table I presents examples of the Freesound raw descriptions and LLM-processed captions. Our fine-tuned T5 model is capable of automatically translating Spanish to English, eliminating

[3][Online]. Available: www.freesound.org
[4][Online]. Available: www.epidemicsound.com
[5]Model name: gpt-3.5-turbo-0301

TABLE I
EXAMPLES OF RAW DESCRIPTIONS AND CLEANED DESCRIPTIONS FROM OUR FINE-TUNED T5-XXL MODEL

Freesound raw description input	LLM filtered output
Title: 06_beso.wav Description: sonido de beso	Sound of a kiss.
Title: nice wave.WAV Description: "i was fiddling around with an analog synth that went into a peavey km50 keyboard amp and i made this sound it has loads of pitch editing giving it the house-ey feel to it and i think it was an fmaj5/9 chord with extra 6ths and 4ths and this one is just a bit longer than the other one"	Analog synth sound with pitch editing creating a house-ey feel. Fmaj5/9 chord with additions of 6ths and 4ths.
Title:Bike04.wav Description:recorded with H4N.	Sound of a motorbike.
Title: "Temporal process.mp3" Description: Dr. Wineski of Morehouse school of Medicine reading a list of anatomical vocabulary words aloud.	Dr. Wineski reading a list of anatomic vocabulary words aloud.

Blue fonts represent successful examples, while red fonts represent failure modes.

redundant or irrelevant details, and summarizing long sentences into one-sentence high-level audio captions. However, the model inaccurately represents the audio recordings in a small portion of the captions. For instance, some captions mistakenly identify 'bike' as 'motorbike' or fail to entirely remove entities, as exemplified at the bottom of Table I.

C. Weakly-Labeled or Unlabeled Datasets

Lastly, the datasets with the largest scale are those that are weakly-labeled or unlabeled, including AudioSet [18] and ACAV100M [34]. Both AudioSet and ACAV100M are comprised of audio clips extracted from YouTube videos, offering a vast range of audio data. AudioSet is a weakly-labeled dataset with 527 predefined sound classes of 2 million 10-second recordings, where each audio clip is marked only by the presence of sound event tags, without detailed descriptions. ACAV100M consists of 100 million videos that exhibit high audio-visual correspondence, yet it lacks of any form of labeling. The ACAV dataset also contains a variety of scales, ranging from 20K to 100M. In this study, we concentrate on ACAV2M, leaving the exploration of larger scales for future work. To leverage these datasets and to accurately represent audio content, we employ an off-the-shelf audio captioning model HTSAT-BART pretrained on WavCaps and fine-tuned on AudioCaps.[6] Specifically, we choose this model because it achieves state-of-the-art performance on audio captioning tasks and is trained on a diverse range of data. Employing the chosen captioning model for synthetic caption generation differs from previous work such as BLAT [35], whose captioner is initially trained on the smaller-scale AudioCaps dataset.

III. NEURAL ARCHITECTURE

We train Cacophony in two stages, as depicted in Fig. 3. First, we train an audio encoder with the MAE objective, using audio-only data from our collected large-scale dataset. Second, we take this trained audio encoder and use it as the initilzation for training the audio-text model with both contrastive and captioning objectives.

[6][Online]. Available: https://github.com/XinhaoMei/WavCaps

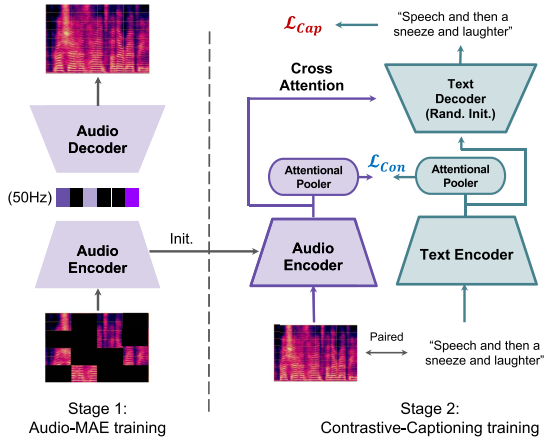

Fig. 3. The proposed system consists of a two-stage training process, as illustrated in the system diagram: Left (Stage 1): Audio-MAE training: This stage is conducted on our collected dataset. After this stage, the audio decoder is discarded, and the encoder is retained to initialize the audio encoder for the second stage of training. Right (Stage 2): Contrastive-captioning training: This stage involves training three components – the audio encoder, the text encoder, and the text decoder. The second stage is dedicated to achieving a contrastive-captioning objective, aligning and fine-tuning the interaction between the audio and text components.

A. Stage 1: Audio Encoder Training

We train our audio encoder with an MAE objective, which involves masking random patches of the input signals (e.g., images or spectrograms) and then reconstructing these masked portions [24], [25]. Specifically, in the MAE training, a transformer encoder initially processes the unmasked patches. Following this, a transformer decoder, which will be discarded before the second-stage contrastive-captioning training, predicts both the masked and unmasked regions. This MAE objective encourages the learning of global, contextualized representations over arbitrary subsets of spectrogram patches [25]. Compared to methods based on supervised classification objectives, MAE does not required labeled data. It also demonstrates improved performance with increased model size [24] and increased dataset size [36].

We initially segment the mel-spectrograms into non-overlapping regular grid patches, following Huang et al. [25], who observe that utilizing overlapping patches leads to worse performance given a fixed compute budget. Subsequently, these patches are flattened, transformed via a learned linear projection, and added with positional embeddings to preserve their time-frequency "spatial" context. Specifically, we employ 1-D fixed sinusoidal positional embeddings along the time axis and learnable positional embeddings along the frequency axis. Then, 80% of the spectrogram patches are randomly masked and discarded prior to encoding. We chose this high masking ratio based on AudioMAE [25], which finds that masking 80% unstructured patches achieves the highest performance on downstream classification tasks. The unmasked spectrogram patches are then processed by the transformer encoder. To obtain an input sequence for the decoder, the encoder output is concatenated sequence-wise with learnable embeddings that represent the masked patches. Since AudioMAE [25] demonstrates that there is no substantial performance improvement using local attention

over global attention in the decoder, we maintain the original vision transformer (ViT) backbone for simplicity. Eventually, a final linear projection layer is used to reconstruct the spectrogram patches. Following [24], [25], the training objective is to minimize the Mean Squared Error (MSE) between the per-patch normalized values of the reconstructed and the original input spectrogram patches. Our preliminary experiments also show that using per-patch normalization achieves better performance in downstream tasks. Finally, we discard the decoder and only keep the trained encoder for the second-stage training.

We elect not to use the Swin-Transformer [37] used by LAION-CLAP [17], since pyramidal ViTs introduce patch merging as well as operations within "local" windows, which would make it difficult to directly handle the random sequence of partial spectrogram tokens [38]. Aside from the issue of incompatibility with MAE, the patch-merging in Swin-Transformer results in significant time decimation, with an output resolution of 6 Hz compared to 50 Hz output of ViT. By avoiding decimation in the time dimension, we can readily adapt our pretrained audio encoder for tasks such as sound event detection with high time-resolution.

B. Stage 2: Contrastive-Captioning Training

Given the audio encoder trained in the first stage, we now introduce a text encoder/decoder and train a contrastive model with an auxiliary captioining objective, in a setup similar to BLIP [27] and CoCa [26]. For the contrastive objective [14], [39], we use learned linear projections to map the text and audio embeddings to the same dimension, followed by an l_2-normalization layer. Then, we use matched audio-text embeddings as positives and all non-paired examples as hard negatives, resulting in N total samples for the pairwise Information Noise-Contrastive Estimation (InfoNCE) loss as described in [40]:

$$\mathcal{L}_{\text{Con}} = -\frac{1}{N}\left(\underbrace{\sum_i \log \frac{\exp(x_i^\top y_i/\tau)}{\sum_{j=1}\exp(x_i^\top y_j/\tau)}}_{\text{audio-to-text}} + \underbrace{\sum_i \log \frac{\exp(y_i^\top x_i/\tau)}{\sum_{j=1}\exp(y_i^\top x_j/\tau)}}_{\text{text-to-audio}}\right), \tag{1}$$

where x_i and y_j are l_2-normalized embeddings of the audio in the i-th pair and text in the j-th pair, respectively, and τ is a learnable temperature.

For the captioning objective, the model is required to autoregressively predict the tokenized text associated with a given audio sample. The text decoder is trained to minimize the negative log-likelihood of current ground-truth token given previous ground-truth tokens:

$$\mathcal{L}_{\text{Cap}} = -\frac{1}{T}\sum_{t=1}^{T} \log P_\theta(y_t|y_{1:t-1}, x), \tag{2}$$

where y_t is the t-th ground-truth token for a given caption y and T is the caption's total length. As a result, we apply both contrastive and generative objectives in the second stage training as follows:

$$\mathcal{L}_{\text{II}} = \mathcal{L}_{\text{Con}} + \lambda_{\text{Cap}} \cdot \mathcal{L}_{\text{Cap}}, \tag{3}$$

where $\lambda_{\text{Cap}} > 0$ is a hyperparameter controlling the relative weight of the captioning loss compared to the contrastive loss.

Architecturally, the second-stage training involves training three key modules: audio encoder, text encoder, and text decoder. For the audio encoder, we use the encoder from the first stage of AudioMAE training, outlined in Section III-A. Our text encoder is a transformer that follows RoBERTa's architecture. However, unlike RoBERTa, we apply causal self-attention rather than bidirectional self-attention. This is a necessary modification to prevent information leakage to the text decoder. We initialize our model with the RoBERTa pretrained weights. The outputs from audio encoder and text encoder are framewise and token-wise embeddings respectively; in order to obtain a single embedding vector for each modality, we choose to integrate multi-head attention poolers on top of the sequential embeddings [26], [41] for the contrastive objective, shown in the right of Fig. 3. Our text decoder consists of stacked transformer layers on top of our text encoder with causal self-attention to prevent next-token prediction leakage. Each transformer block has a cross-attention layer that attends to the full output of our audio encoder. The text decoder layers are initialized randomly.

Because our audio data used for training varies in length, we follow a similar training strategy as the first MAE training stage: for audio shorter than the training length, we utilize zero-padding and generate corresponding masks; for audio exceeding the training length, we randomly sample time-frequency patches from the full spectrograms and feed them into the audio encoder. It is worth noting that for shorter audio, this differs from the first-stage training: During MAE training, 80% of spectrogram patches are masked, whereas during contrastive training, audio shorter than the predefined training length is fully unmasked.

IV. Implementation Details

A. Training and Inference Setup

As outlined in Section II, the training datasets are categorized into three types: the 'clean-labeled' dataset consists of 1,212 hours, the 'noisy labeled' dataset consists of 3,003 hours, and the 'weakly/unlabeled' dataset consists of 9,017 hours. All audio files are processed into a mono channel with a sampling rate of 16 kHz. We extract mel-spectrograms using a 25 ms window and a 10 ms hop length, with 128 mel bands and 512 as FFT size. The spectrograms are then converted into 16×16 patches without overlap.

In the first-stage training, we randomly sample 15 seconds of audio for each recording, corresponding to a patch sequence length of 750. At the second stage of training, we limit the audio length to 10.24 seconds, equivalent to a patch sequence length of 512, i.e., when the patch sequence length exceeds 512, we randomly sample 512 patches from the full sequence, following Patchout faSt Spectrogram Transformer (PaSST) [42]. Text

samples are tokenized with the pretrained RoBERTa tokenizer and truncated to maximum length of 77. Our audio encoder, employed in both training stages, is a 12-layer ViT-Base (ViT-B) Transformer with 8 attention heads, a hidden size of 768 and an intermediate size of 3072, and utilizes the SiLU activation function. Additionally, a 12-layer decoder with the same settings is used during the MAE training phase. For the attention pooler, we choose to use $n_{head} = 8$.

When training AudioMAE, we use a batch size of 512 and a learning rate of 2×10^{-4}, and employ the Adam with decoupled Weight decay (AdamW) [43] optimizer with a weight decay of 0.01. We train for 200,000 steps, with a 10,000 step learning rate warm-up and a cosine decay to 1×10^{-6}. In the contrastive stage, we set $\lambda_{\text{Cap}} = 1$ as the default value. While fine-tuning this parameter may marginally improve performance, as suggested by Yu et al. [26], we find the default setting to be sufficiently effective for our purposes. During training, we again use AdamW with weight decay 0.01 with a batch size of 4096. The learning rate is warmed up for 10,000 to a peak of 1×10^{-5} and is decayed to 1×10^{-6} over 300,000 (total) steps following a cosine schedule.

During inference, we process the full audio sequence if it fits in the memory. For longer sequences exceeding memory capacity, we randomly sample a subset of spectrogram patches that fit within available memory. This approach allows us to handle variable-length inputs while adapting to computational limitations. Notably, despite the potential for duration distribution shift between training (where we use masked or dropped spectrogram patches) and inference (where we process full sequences), we observe improved performance when utilizing all available audio information during inference, aligning with findings in [23].

B. Sharpness-Aware Minimization

Given our use of a large training batch size of 4096 for the contrastive objective and only having nearly 4 million pairs of audio-text data, one epoch is completed in only around 1,000 steps. When training our contrastive model initially, we empirically observe overfitting early in the training process. DALLÂ·E-2 [44] found that using Sharpness-Aware Minimization (SAM) [45] in training CLIP models improves performance; we thus investigate using SAM in our contrastive training stage. SAM is an optimization technique that encourages convergence toward flatter local minima with the hope of improving model generalizability, which is also shown to provide robustness against label noise [45].

More concretely, we denote $S = (\boldsymbol{x}_i, y_i)_{i=1}^{n}$ as the training dataset and $\boldsymbol{w}$ as the trainable parameters. SAM seeks to find the parameters whose neighbors have low training loss through the following objective:

$$\min_{\boldsymbol{w}} \mathcal{L}_S^{SAM}(\boldsymbol{w}) = \min_{\boldsymbol{w}} \max_{\|\boldsymbol{\epsilon}\|_p \leq \rho} \mathcal{L}_S(\boldsymbol{w} + \boldsymbol{\epsilon}), \tag{4}$$

where $p \geq 1$ defines the order of the norm and $\rho \geq 0$ is the tunable hyperparameter that measures the size of the neighborhood, and $\mathcal{L}_S$ can be any arbitrary loss function on dataset S. The inner maximum operator computes the maximum loss within the neighborhood. Its difference from $\mathcal{L}_S(\boldsymbol{w})$ defines how quickly the loss increases from $\boldsymbol{w}$, i.e., the sharpness of the loss landscape. The SAM loss is thus minimizing not only the $\mathcal{L}_S(\boldsymbol{w})$ itself, but also the sharpness within the neighborhood. To solve the inner maximization problem efficiently, we can use the first-order Taylor expansion approximation, i.e., $\mathcal{L}_S(\boldsymbol{w} + \boldsymbol{\epsilon}) \approx \mathcal{L}_S(\boldsymbol{w}) + \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{w}} \mathcal{L}_S(\boldsymbol{w})$ [45]. With $p = 2$, the inner maximization can be achieved at:

$$\hat{\boldsymbol{\epsilon}}(\boldsymbol{w}) \approx \arg\max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} \epsilon^T \nabla_{\boldsymbol{w}} \mathcal{L}_S(\boldsymbol{w}) = \rho \frac{\nabla_{\boldsymbol{w}} \mathcal{L}_S(\boldsymbol{w})}{||\nabla_{\boldsymbol{w}} \mathcal{L}_S(\boldsymbol{w})||}, \tag{5}$$

which is a scaled version of the gradient, normalized to have a magnitude of ρ. Substituting $\hat{\boldsymbol{\epsilon}}$ back into (4) and differentiating it w.r.t. $\boldsymbol{w}$, we obtain:

$$\nabla_{\boldsymbol{w}} \mathcal{L}_S^{SAM}(\boldsymbol{w}) \approx \nabla_{\boldsymbol{w}} \mathcal{L}_S(\boldsymbol{w} + \hat{\boldsymbol{\epsilon}}(\boldsymbol{w})) \approx \nabla_{\boldsymbol{w}} \mathcal{L}_S(\boldsymbol{w})|_{(\boldsymbol{w}+\hat{\boldsymbol{\epsilon}})}, \tag{6}$$

where the first approximation is due to the Taylor expansion mentioned before, while the second approximation is due the removal of a higher-order differentiation term. Note that two forward/backward passes are required to compute every update: one for computing the perturbation $\hat{\boldsymbol{\epsilon}}$ and the other one for computing the gradient at the perturbed weight vector $\boldsymbol{w} + \hat{\boldsymbol{\epsilon}}$; thus, the wall-clock time per step is essentially doubled. There has been recent work [46] in reducing SAM's computational overhead; we leave the application of these techniques to future work.

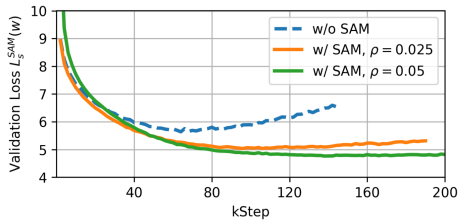

Fig. 4. Contrastive-Captioning objective on the validation set during training, comparing scenarios with and without the application of SAM. For the cases with SAM optimization, we employ various neighborhood sizes as determined by the hyperparameter ρ.

The training dynamics, depicted in Fig. 4, show the impact of incrementally adjusting ρ by increments of 0.025. Increasing ρ corresponds to a larger $\boldsymbol{\epsilon}$ perturbation and thus heavier regularization. The figure clearly shows that without SAM, the model is prone to overfitting on the training data as early as 40,000 steps. In contrast, with even a small degree of sharpness minimization at $\rho = 0.025$, overfitting is delayed until approximately 100,000 steps and occurs at a significantly lower objective value. In our final second-stage model training, we use SAM with $\rho = 0.075$.

V. Evaluation

A. Evaluation Overview

By employing both contrastive and captioning training objectives, Cacophony is capable of providing audio-text representations and generating free-form text for open-ended audio

understanding tasks, whereas the vanilla CLAP models cannot generate text without separate adapter modules.

1) Tasks: We evaluate Cacophony's performance across three primary categories: (1) Audio-Text Representation Tasks: These include audio-text retrieval, closed-ended audio question answering, and zero-shot transfer in audio classification. These assessments concentrate on the model's performance in audio-text cross-modal alignment. (2) General-Purpose Audio Representation Tasks: To evaluate Cacophony's pretrained audio encoder as a general-purpose audio representation model, we conduct audio classification and sound event detection tasks. For these assessments, we train a Multi-Layer Perceptron (MLP) on top of our frozen audio encoder. (3) Open-Ended Audio Understanding Task: We assess our model's capacity of generating free-form text through audio captioning. However, we have excluded open-ended audio question answering from our evaluation, as it requires additional text pre-processing and fine-tuning steps beyond the scope of this study.

2) Evaluation Datasets: In the audio research community, some major high-impact datasets originate from Freesound, including Clotho [13], ESC50 [47], UrbanSound8K [48], and FSD50K [49]. Another set of popular datasets comes from YouTube, such as Audioset (and its subset AudioCaps) and VGGSound [50]. Given that our dataset includes Freesound and 4 million Youtube samples, we need to ensure that none of the data on which we evaluate is present in our train dataset. FSD50K, a subset of Freesound, is originally in the HEAR benchmark. Because FSD50K is of high-quality and is relatively large, rather than removing FSD50K from our training dataset, we exclude it from our HEAR benchmark evaluation. *Mridingham Stroke and Mridingham Tonic* also overlaps with Freesound; we choose to exclude it from the HEAR benchmark for simplicity, as it is small-scale relative to other HEAR tasks. Lastly, we exclude *Beehive* tasks from our HEAR evaluation because they require inference on 600-second long samples, and the models to which we compare are unable to process audio of this length.

B. Audio-Language Retrieval

1) Task Definition: The audio-text retrieval task involves searching for a specific audio clip or a caption based on a query from the other modality. Text-to-audio retrieval involves retrieving audio for a given text caption, and audio-to-text retrieval involves retrieving text for a given audio sample.

2) Experimental Setup: To effectively perform retrieval tasks, pretrained contrastive models are used to predict if a given audio clip and text description are paired together. Following prior works [14], [17], [51], for audio-to-text retrieval, we first compute the feature embeddings for the target audio clip and the corresponding set of potential text captions. Subsequently, we compute the cosine similarity between the audio embedding and every text embedding. The retrieved text samples are then selected based on the highest cosine similarity scores (the text-to-audio retrieval process is identical with the roles reversed). During evaluation, we benchmark the audio-text retrieval task on the test splits of AudioCaps and Clotho datasets. We use recall at rank k ($R@k$) as our evaluation metric. For a given query, $R@k$ is assigned a value of 1 if the relevant item is among the top k retrieved items and 0 if it is not. This $R@k$ score is then averaged across the entire dataset to obtain the final performance.

3) Result: The audio-language retrieval results on the AudioCaps and Clotho datasets are presented in Table II, where we compare against the most recent contrastive-based models, including MS-CLAP [32], LAION-CLAP [17], WavCaps [19] and FLAP [22]. To ensure a fair comparison, we only compare with pretrained models, rather than models that are fine-tuned for specific audio-text retrieval tasks.

Our proposed method has achieved state-of-the-art or comparable performance to the best-performing systems across all evaluated metrics and both datasets. Compared to WavCaps and LAION, our model achieves better performance on AudioCaps and comparable results on Clotho with 'WavCaps-CNN14'. When compared against FLAP, Cacophony outperforms the non-feature-fusion-based model and matches the fusion-based model. While both Cacophony and FLAP share some common elements, such as LLM-augmented captions and MAE-based audio encoders, our model's superior performance can be attributed to our two-stage MAE and contrastive training strategy, together with SAM optimization techniques. This combination allows for more effective learning of audio-text representations, resulting in improved retrieval performance across both datasets.

C. Closed-Ended Audio Question Answering

1) Task Definition: AQA is the task of generating a text response to a text question about an audio signal. We follow the strategy in [29] to cast AQA into a supervised classification task with a closed-ended answer set.

2) Experimental Setup: We evaluate our approach on two datasets: Clotho-AQA [29] and Music-AQA [53]. Clotho-AQA contains 1,991 audio samples, each with six questions. Four questions have yes/no answers, while two have single-word answers. Music-AQA, derived from the Music-AVQA dataset, includes 6,319 one-minute audio clips from musical performances. Each clip has one question assessing counting or comparison skills. Both datasets feature binary or single-word answers, with Music-AQA responses ranging from 'zero' to 'nine'. We evaluate closed-ended AQA through training small MLPs on top of frozen pretrained encoders. In particular, we first extract embeddings from the frozen audio and text encoders from the contrastive models, and then we concatenate them into one vector and pass it through a 4-layer MLP for binary or multi-class classification.

In addition to several audio-text models, we also compare our approach to a baseline model [52] designed specifically for AQA that achieves state-of-the-art performance on the Clotho-AQA and Music-AQA benchmarks. This model differs from contrastive models in two ways: First, its audio and text encoders are pretrained independently, and second, after extracting frame-level audio and token-level word embeddings, these embeddings are processed through fused attention layers followed by MLP layers. We follow the train, validation and test splits provided in the original Clotho-AQA and Music-AQA datasets. We use $R@k$ to evaluate the performance on the Clotho-AQA dataset

TABLE II
SYSTEM COMPARISONS ON AUDIO-LANGUAGE RETRIEVAL ON TEST SETS OF AUDIOCAPS AND CLOTHO

Model	AudioCaps						Clotho					
	Text-to-Audio			Audio-to-Text			Text-to-Audio			Audio-to-Text		
	R@1	*R@5*	*R@10*	*R@1*	*R@5*	*R@10*	*R@1*	*R@5*	*R@10*	*R@1*	*R@5*	*R@10*
CLAP-HTSAT [32]	34.6	70.2	82.0	41.9	73.1	84.6	16.7	41.1	54.1	20.0	44.9	58.7
LAION [17]	36.1	71.8	83.9	46.8	82.9	90.7	16.1	38.3	51.1	22.7	48.5	60.8
LAION (fusion) [17]	35.1	71.5	83.6	45.8	80.9	91.6	18.2	42.5	54.4	25.7	51.5	63.4
WavCaps-CNN14 [19]	34.7	69.1	82.5	44.6	76.3	86.2	**21.2**	46.4	**59.4**	25.9	52.6	65.8
WavCaps-HTSAT [19]	39.7	74.5	86.1	51.7	82.3	90.6	19.5	45.2	58.2	23.4	50.9	63.4
FLAP [22]	40.4	74.7	85.0	51.5	82.5	92.5	17.4	41.3	53.7	21.6	51.2	63.1
FLAP (fusion) [22]	**41.5**	**75.5**	86.0	53.0	**84.1**	**92.6**	20.3	**46.5**	58.8	25.5	53.4	**67.9**
Cacophony (ours)	41.0	75.3	**86.4**	**55.3**	83.6	92.4	20.2	45.9	58.8	**26.5**	**54.1**	67.3

Results for the baselines are copied from the references. *R@k* (Recall at *k*) represents the proportion of queries for which the relevant item is retrieved within the top k results. Higher values indicate better performance.

TABLE III
EVALUATION OF AUDIO QUESTION ANSWERING IN CLOTHO-AQA AND MUSIC-AQA BENCHMARKS

Recall (%)	Clotho-Word			Clotho-Bin.	Music-Word	Music-Bin.
	R@1	*R@5*	*R@10*	Acc.	Acc.	Acc.
MWAFM [52]	**21.3**	**45.5**	**56.7**	68.6	58.0	70.5
LAION	17.8	42.1	53.2	68.4	**58.9**	69.2
LAION (fusion)	18.7	42.4	53.3	68.2	57.4	71.2
MS-CLAP	19.4	43.7	54.5	68.8	53.7	74.5
WavCaps-CNN14	**20.3**	**44.3**	**55.5**	66.8	54.3	71.1
WavCaps-HTSAT	18.1	41.5	53.0	68.4	53.3	74.2
Cacophony (Ours)	19.7	42.6	52.0	**70.7**	53.6	**74.9**

("Word" stands for singleword closed-vocabulary classification, "Bin." stands for binary classification. "Acc." stands for accuracy).

TABLE IV
SYSTEM COMPARISONS ON ZERO-SHOT CLASSIFICATION. EACH DATASET IS MARKED WITH THE ORIGINAL SAMPLING RATE

Accuracy (%)	VGGSound* (48k)	TUT-AS (44.1k)	ESC-50 (44.1k)	US-8K (44.1k)
MSCLAP [32]	12.3	25.1	81.6	73.1
LAION [17]	**30.0**	47.4	84.4	76.1
LAION (fusion) [17]	26.9	27.5	84.1	71.6
WavCaps-CNN14 [19]	27.5	47.2	87.0	72.9
WavCaps-HTSAT [19]	28.9	**49.4**	93.1	**80.4**
Cacophony (ours)	27.1	48.6	**93.4**	77.1

*:Due to the inaccessibility of many YouTube videos, our VGGSound test split comprises a total of 12,722 samples.

because the size of the answer vocabulary is large (828). We use accuracy to evaluate the performance on the Music-AQA dataset because its single-word answer vocabulary consists of only 10 possible answers to different questions.

3) Result: In our evaluations, detailed in Table III, we benchmark fine-tuned contrastive models together with a state-of-the-art AQA baseline. Although Cacophony outperforms other contrastive models in binary question tasks, it performs less well in single-word classification compared to the specialized AQA baseline. We believe this is because Cacophony uses pooled embeddings from a frozen text encoder, while the baseline learns to aggregate over granular text features [52]. As the text in AQA task, *i.e.*, a question, is out-of-distribution for our text encoder, the frozen pooled embeddings may not precisely capture the question's semantics; A trainable pooler could offer improved pooled embeddings by leveraging the fine-grained output of the encoder.

D. Zero-Shot Transfer on Audio Classification

1) Task Definition: Audio classification is to categorize audio recordings into different sound types. To evaluate the generalization capability of the contrastive models on unseen datasets, we explore their zero-shot transfer ability on these audio classification tasks following previous practices [14], [17], [51]. This setup resembles the audio-to-text retrieval task defined in Section V-B, with the key difference being the use of a predefined set of labels as text descriptions.

2) Experimental Setup: We evaluate the models' zero-shot classification accuracy using four datasets: VGGSound-test, TUT Acoustic Scenes [54], ESC-50 [47], and UrbanSound8K [48]. These datasets collectively cover a wide range of sound events. For each dataset during classification, we use the names of all classes within that dataset as the set of possible text descriptions, and as in audio-to-text retrieval, select the matching text based on cosine similarity. Since the text descriptions seen during training are relatively verbose compared to the often-single-word labels in classification tasks, we craft individual prompt templates for each dataset. For instance, in the context of sound event classification, we use a template such as "This is a sound of [label]", while for acoustic scene classification, we use "This sound is on [label]." We use Top-1 accuracy as the evaluation metric.

3) Result: We compare our model, Cacophony, against all baseline models on zero-shot classification on a variety of audio classification benchmarks, presented in Table IV. Overall, Cacophony exhibits competitive performance for general sound categories on ESC-50, UrbanSound8K, and TUT-Acoustic Scenes. However, it achieves a lower accuracy on the VGGSound dataset. This relatively lower performance on VGGSound, which includes 310 classes, suggests that its granularity may be too fine for the model to differentiate different classes. This challenge is likely due to the presence of inaccuracies within the re-captioned audio descriptions in the training data, potentially resulting from both the LLM cleaning and automatic captioning processes. Additionally, since our model is trained on audio sampled at 16,000 Hz, training on full-bandwidth audio recordings could potentially improve its performance in zero-shot classification tasks, as demonstrated in previous work [2].

E. Holistic Evaluation of Audio Representations

1) Task Definition: The HEAR benchmark is designed to evaluate the effectiveness of general audio representations across various domains, including speech, music and general sound. The HEAR benchmark comprises two principal task types: (1)

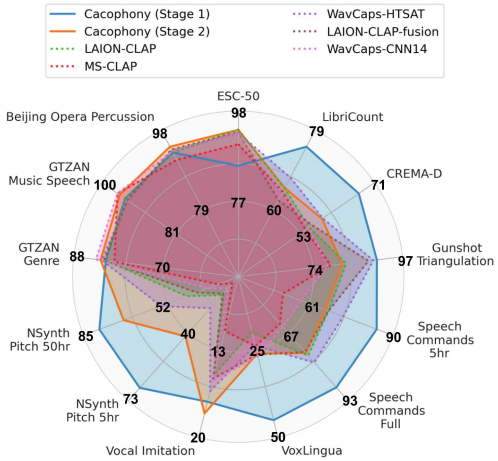

Fig. 5. Comparison of classification accuracy (%) on HEAR benchmark. Evaluation scores are stable across tasks, with a median 95% confidence interval of 0.25% with `hear-eval-kit` model-selection strategy. Comparisons of audio branches from contrastive language-audio embeddings.

scene-based tasks, which involve classifying an entire audio clip, and (2) event-based tasks, which aim at identifying specific sound events over time, i.e., predicting the start time, end time, and label for each sound event.

2) Experimental Setup: We follow the fine-tuning strategies presented in [30] using `hear-eval-kit`.[7] Specifically, in both scene-based and event-based tasks, the audio encoders from the audio-text models are frozen and used as the input feature vector to a shallow downstream MLP classifier. During evaluation following [30], scene-based tasks are measured with classification accuracy. Event-based tasks are evaluated with event onset F-measure, which correlates better with human perceptual than framewise classification accuracy [58].

3) Result: We evaluate our models against existing state-of-the-art contrastive models using the HEAR benchmark, shown in Fig. 5. Our proposed system demonstrates good performance across various tasks, outperforming or closely matching existing approaches. Notably, it achieves significantly better accuracy than other contrastive models in the pitch classification tasks defined in *NSynth Pitch*. WavCaps-HTSAT and our Cacophony models show relatively balanced performance across different datasets but no single model is superior across all tasks. Cacophony underperforms WavCaps-HTSAT baseline in several speech-related datasets, including *Speech Commands* and *LibriCount*. We believe that incorporating more speech-specific data would improve performance on these speech-related tasks.

Interestingly, in a comparison between Cacophony (stage 1) and Cacophony (stage 2), i.e., before and after the contrastive-captioning training, we notice a significant drop in performance across various tasks. This suggests that the contrastive training objective, which encourages the audio encoder to extract features relevant to selecting a text pair, may not align with other audio classification objectives.

For event-based tasks, we do not compare our model with existing contrastive models, because they tend to perform poorly without manually computing shifting-windowed embeddings, as they typically use temporal pooling in convolutional blocks or downsampling in Swin blocks [37]. Instead, we compare our model against top-performing individual models (i.e., excluding systems that ensemble multiple models) on the sound event detection tasks listed on the HEAR leaderboard.[8] We exclude ensemble systems for a fair comparison on individual model performance.

TABLE V
COMPARISON WITH TOP-PERFORMING NON-ENSEMBLE BASED SYSTEMS ON EVENT-BASED TASKS IN HEAR BENCHMARK LEADERBOARD USING ONSET F1 SCORE (%)

Event Onset F-1 (%)	DCASE2016 T2	MAESTRO 5h
OpenL3 [55]	**83.2**	1.7
wav2vec2 [56]	66.3	3.3
SONY-ViT	66.8	23.9
CREPE [57]	50.4	**40.1**
Cacophony (Ours)	81.1	10.0

Results from other systems are taken directly from the HEAR Leadboard.

Table V shows the results. Our model demonstrates competitive performance in the "DCASE 2016 task 2" with a score of 81.1%, which is among the top-performing systems. However, its capability appears more limited in the "MAESTRO 5h" task, scoring only 10.0%. This suggests that it may not be as effective in tasks that require fine-grained instrumental pitch detection, in contrast to daily environmental sounds.

F. Automated Audio Captioning

1) Task Definition: Automated audio captioning involves generating a free-form textual description for an audio signal, moving beyond predefined class labels or tags.

2) Experimental Setup: At inference, our model utilizes temperature sampling to generate captions at a temperature of 0.1. The baselines apply beam search with a beam size of 3 for generating text. We evaluate on the AudioCaps and Clotho datasets. For evaluation metrics, we use the Microsoft COCO Caption Evaluation package [59], which includes BLEU_n, ROUGE_L, METEOR, CIDEr, SPICE, and SPIDEr.

3) Result: We benchmark our captioning head against HTSAT-BART and CNN14-BART baselines proposed in [19] which are pretrained on WavCaps. Both of these models have shown good performance in the automated audio captioning tasks on the Clotho and AudioCaps datasets. We also include the synthetic audio captioner introduced in Section II, HTSAT-BART-FT, which has been additionally fine-tuned on AudioCaps.

The results of this comparison are detailed in Table VI. In the AudioCaps dataset, our model exhibits competitive performance with that of the baseline models. However, there is still a noticeable performance gap when compared to our "teacher model" HTSAT-BART-FT. On the Clotho dataset, Cacophony's performance is notably lower than both CNN14-BART and HTSAT-BART. In Table VII, we include a few captiong examples of Cacophony that received low scores. Looking at these examples, we identify two primary causes for our model's weak

[7][Online]. Available: https://github.com/hearbenchmark/hear-eval-kit

[8]HEAR leaderboard (https://hearbenchmark.com/hear-leaderboard.html)

TABLE VI
AUTOMATED AUDIO CAPTIONING RESULTS ON TEST SETS OF AUDIOCAPS AND CLOTHO

Model	$BLEU_1$	$BLEU_4$	$ROUGE_l$	METEOR	CIDER	SPICE	SPIDEr
AudioCaps							
HTSAT-BART-FT	**70.7**	**28.3**	**50.7**	**25.0**	**78.7**	**18.2**	**48.5**
CNN14-BART	67.0	26.1	48.3	23.1	72.1	16.9	44.5
HTSAT-BART	67.5	27.2	48.3	23.7	71.1	17.7	44.4
Cacophony (Ours)	68.4	25.9	48.6	23.6	72.8	16.8	44.8
Clotho							
CNN14-BART	56.0	16.0	37.0	17.1	39.3	11.7	25.5
HTSAT-BART	**57.6**	**16.4**	**38.2**	**17.5**	**41.5**	**11.9**	**26.7**
Cacophony (Ours)	50.8	11.5	34.3	15.3	34.2	10.6	22.4

All compared baselines are from WavCaps [19], the evaluation results for the baselines are from the reference. HTSAT-BART-FT is fine-tuned specifically on AudioCaps, and we also used it for recaptioning weakly/unlabeled datasets.

TABLE VII
REPRESENTATIVE EXAMPLES OF AUTOMATIC CAPTIONING BY CACOPHONY ON CLOTHO DATASET

Ground Truth	Generated by Cacophony
Plastic and other materials rustle and crinkle continuously.	A sound of a plastic bag being crumpled.
A quiet environment with a few insects making a sound and some birds chirping far away.	Crickets chirping in the background.
A door is being unlatched creaking open and being fastened again.	A door opening and closing.

performance: First, the textual output from Cacophony seems to show a different writing style from the ground-truths, lacking some detailed descriptions that are characteristic of Clotho's captions. It is expected that using a captioning model fine-tuned on Clotho to generate our pretraining data would enhance Cacophony's performance on this dataset, albeit potentially at the expense of performance on AudioCaps. Second, Cacophony's audio encoder does not successfully detect all sound events. This is evidenced in the middle example, where the "birds chirping far away" event is not correctly identified by our model.

G. Modality Gap

Although the contrastive training objective brings matched text-audio pairs closer than non-matched pairs, it does not necessarily bring audio and text into a joint multi-modal space globally. Liang et al. [60] define the modality gap as the difference between the centroids of audio and text embeddings: $\Delta_{gap} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i - \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i$, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ denote the L2-normalized audio and text embeddings for the i -th sample in a given dataset. The magnitude of this gap provides a quantitative measure of the global discrepancy between the audio and text embeddings. This gap is shown to arise from the general inductive bias of deep neural architectures, random initialization and training objective [60]. Another factor may be the noisy correspondence of audio-text pairs, since correctly matched pairs guide the training, while mismatched pairs provide misleading supervision, where the pairs are incorrectly matched, as pointed out in Luong et al. [61], where the correctly matched pairs guide the training, the noisy pairs incorrectly supervise the training. As shown in contrastive image-text models, the existence of modality gap can impact its transferability to downstream tasks [60]; we hypothesize that it may also be the case for contrastive audio-text models. In our experiment, we track the evolution of the modality gap by plotting paired text and audio embeddings, as illustrated in Fig. 6(a). We observe that the modality gap begins to increase once the validation loss plateaus after 120k iterations. When evaluating the text-audio retrieval task on the validation split, we find that checkpoints with smaller modality gap do not necessarily guarantee significantly better performance. However, a model with a narrower gap may offer advantages in applications where embedding replacement across modalities is required, such as in text-to-audio generation [8], [9] or language-guided source separation [10], [11]. We also visualize the modality gaps of existing contrastive models, as shown in Fig. 6(b). In models like LAION-CLAP and WavCaps-HTSAT, we notice that text and audio embeddings form distinct clusters.

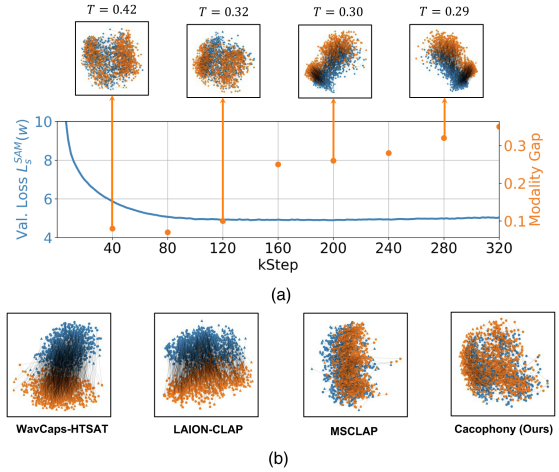

Fig. 6. UMAP visualization of text and audio embeddings on the validation set. (a) Evolution of modality gap during our second-stage training, T stands for the temperature parameter in (1). (b) Visualization of modality gap of existing models.

VI. ABLATION STUDIES

In this section, we use audio-text retrieval for the ablation studies, as the task aligns with our pretraining contrastive objective. In each ablation study, we use a 12-layer ViT-Base (ViT-B) Transformer with a hidden size of 512 and intermediate size of 1024 as our audio encoder in both training stages. The other training hyper-parameters remain the same as those we described in Section IV. At inference time, we choose to make use of the full sequence length.

A. Text Cleaning

We first explore text pre-processing methods for the contrastive-captioning training stage, using only the Freesound, AudioCaps, and Clotho datasets. To be specific, the compared text pre-processing methods include: (1) implementing LLM cleaning as detailed in Section II-B; (2) conducting recaptioning based solely on audio, as described in Section II-C; (3) directly using raw text within a predefined maximal text length. For recaptioning the Freesound dataset (2), we use CNN-14-BART from WavCaps [19], pretrained on WavCaps and then fine-tuned on Clotho, as its domain matches with Freesound.

The experimental result can be found in Table VIII. When compared to the raw text baseline, LLM cleaning generally improves performance for both text-to-audio and audio-to-text

TABLE VIII
ABLATIONS ON TEXT PREPROCESSING, TRAINED ON OPENSFX, EVALUATED ON TEST SPLIT OF AUDIOCAPS AND CLOTHO

Recall (%)	AudioCaps						Clotho					
	Text-to-Audio			Audio-to-Text			Text-to-Audio			Audio-to-Text		
	$R@1$	$R@5$	$R@10$	$R@1$	$R@5$	$R@10$	$R@1$	$R@5$	$R@10$	$R@1$	$R@5$	$R@10$
None	32.5	67.5	80.1	45.4	**76.5**	86.9	14.2	37.2	51.3	21.2	**51.9**	**66.2**
LLM cleaning	**33.1**	**68.8**	**80.9**	**45.7**	76.1	86.0	17.2	41.2	55.2	**24.8**	51.2	65.8
Recaptioning	32.6	66.8	79.4	42.6	75.6	**88.1**	**17.7**	**43.0**	**56.9**	22.9	51.8	65.4

TABLE IX
ABLATIONS ON THE USE OF DATASET FOR DIFFERENT STAGE TRAINING, V_x-V_y REPRESENTS THAT V_x DATASET IS USED FOR 1ST STAGE MAE TRAINING AND V_y DATASET IS USED FOR 2ND STAGE CAPTIONING CONTRASTIVE TRAINING

Index	**Recall** (%)	AudioCaps						Clotho					
		Text-to-Audio			Audio-to-Text			Text-to-Audio			Audio-to-Text		
		$R@1$	$R@5$	$R@10$	$R@1$	$R@5$	$R@10$	$R@1$	$R@5$	$R@10$	$R@1$	$R@5$	$R@10$
(a)	V_1 - V_1	33.1	68.8	80.9	45.7	76.1	86.0	17.2	41.2	55.2	24.8	51.2	65.8
(b)	V_2 - V_1	32.4	66.9	79.2	42.7	76.0	85.8	15.7	41.0	55.1	21.5	49.5	64.0
(c)	V_3 - V_1	34.5	67.9	80.5	46.0	76.1	86.1	17.0	42.1	56.0	22.7	51.1	65.6
(d)	V_2 - V_2	31.6	64.2	77.7	44.4	78.3	87.7	18.6	42.7	56.8	25.0	52.2	66.3
(e)	V_3 - V_2	**40.5**	**75.6**	86.2	55.5	**84.4**	91.8	19.9	44.3	57.3	24.8	52.2	66.1
(f)	V_3 - V_3	**40.5**	75.2	**86.7**	**55.8**	83.1	**92.1**	**20.1**	**44.6**	**58.0**	**28.8**	**53.9**	**67.9**
(g)	$\emptyset$ - V_3	39.9	74.1	85.7	54.2	**84.3**	92.0	17.2	41.2	54.4	25.0	50.3	65.0

TABLE X
DETAILED INFORMATION OF TRAINING DATASETS IN THE DATASET SCALE ABLATION STUDY

Dataset	Clean Labeled	Noisy-Labeled	Weakly/Un-Labeled	Duration (kHour)
V_1	AudioCaps, Clotho, OpenSFX	Freesound	-	4.2
V_2	AudioCaps, Clotho, OpenSFX	Freesound	ACAV2M	7.8
V_3	AudioCaps, Clotho, OpenSFX	Freesound	AudioSet, ACAV2M	13.2

retrieval tasks on Clotho datasets, and achieves similar performance on AudioCaps dataset. Recaptioning yields mixed results; it improves performance on the Clotho dataset but slightly reduces accuracy on AudioCaps dataset.

For both text preprocessing methods, improvements are more consistent and significant in Clotho than in AudioCaps. We believe that Clotho evaluation is a more reliable indicator of the overall performance than AudioCaps evaluation, as the training dataset used in this ablation aligns closely with Clotho and is out-of-distribution for AudioCaps.

B. Dataset Scale

We perform an ablation study on the effect of training dataset size on model performance in the audio-text retrieval tasks (see Table IX). To do so, we create three datasets designated as V_1, V_2 and V_3 with increasing sizes. Detailed information of these datasets is provided in Table X.

We first study the effect of dataset scale in the first-stage training. In Table IX), each of the three groups, (b-c), (d-e) and (f-g), uses a different dataset in the first-stage training but the same dataset in the second-stage training. In particular, Group (f-g) constitutes the extreme case: (f) uses the entire dataset in the first-stage training and (g) bypasses the first stage entirely. Observations show a performance improvement across all tasks and datasets as the first-stage training data expands. This evidences the positive impact of using larger datasets during the first-training stage on the model's overall effectiveness. There are, however, two exceptions: There is no noticeable improvement in retrieval performance when comparing (a) and (c), and there is a decline in all evaluation metrics when additional data from ACAV2M is incorporated when comparing (a) and (b). This lack of improvement in (b) and (c) compared to (a) may be due to the evaluation dataset being out-of-distribution of the limited training dataset V_1 used in the second stage.

We now analyze effects of data scale in the second-stage training, i.e., contrastive-captioning training. In the comparison of experiments (c), (e), and (f), the audio encoder is initialized with weights from training on the V_3 dataset. During the second stage, these groups are trained on V_1, V_2, and V_3, respectively. When evaluating on AudioCaps, there is a significant increase from (c) to (e) in recall when integrating ACAV2M (V_2) into the second-stage training, but this improvement appears to plateau from (e) to (f) upon the inclusion of AudioSet (V_3). However, on Clotho, the retrieval performance consistently improves with increasing dataset scale from (c) to (e) and (f), without observation of a plateau, showing the effectiveness of increasing data scale.

In experiments (a), (d), and (f), we increase the dataset scale simultaneously for both training stages and observe a gradual improvement on retrieval performance on the Clotho test dataset. However, a significant improvement in retrieval performance on AudioCaps is observed only when integrating AudioSet (group (f)) into both training stages.

We find scaling up the datasets in both stages tends to lead to improved model performance. The MAE training step contributes to the better retrieval performance for the audio-text model. Therefore, collecting more unlabeled audio or using larger versions of the ACAV dataset for audio encoder MAE training could potentially yield further gains in downstream performance.

C. Architectural Design

Regarding the neural network architecture, we examine the effect of incorporating a captioning head, SAM optimization, and fixed length processing window on audio-text retrieval tasks (see Table XI). For these experiments, we use the full-size dataset V_3 in both training stages, i.e., setting (f) in Table IX. We further

TABLE XI
ABLATIONS ON DIFFERENT NETWORK CONFIGURATIONS AND INITIALIZATION

Index	Recall (%)	AudioCaps						Clotho					
		Text-to-Audio			Audio-to-Text			Text-to-Audio			Audio-to-Text		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
(a)	Cacophony	**40.5**	**75.2**	**86.7**	**55.8**	**83.1**	**92.1**	**20.1**	**44.6**	**58.0**	**28.8**	**53.9**	**67.9**
(b)	- *w/o* Captioning head	38.9	73.2	85.4	54.5	82.0	91.3	18.6	42.8	56.6	27.6	52.2	64.5
(c)	- *w/o* SAM	37.1	71.1	83.3	48.7	77.7	88.9	15.5	37.9	51.8	19.6	42.8	55.1
(d)	- *w/* Fixed Length	-	-	-	-	-	-	19.1	43.8	57.4	27.8	52.8	67.1

TABLE XII
CLASSIFICATION ACCURACY (%) ON LONG-SEQUENCE CLASSIFICATION FROM HEAR BENCHMARK

Dataset	Duration (sec)	Fixed Length	Full Length
GTZAN-Genre	30	83.7	85.4
GTZAN-Music Speech	30	97.7	98.5
VoxLingua 107	18.6	25.0	26.5

"Fixed length" refers to randomly sampling a 10-second patch to match the audio duration in training. "Full length" refers to using the full audio.

demonstrate the adaptability of our audio encoder in handling varying audio lengths.

1) Captioning Objective: By comparing experiments (a) and (b) in Table XI, we observe a marginal improvement across both test sets when the captioning objective is included. We hypothesize that the captioning objective facilitates the learning of more fine-grained audio representations, which aligns with the findings of BLIP [27], CoCa [26] and CapCa [62].

2) Use of SAM: By comparing (a) and (c), we see that SAM significantly improves the accuracy on the audio-text retrieval task for both datasets. However, the benefits of SAM come at the expense of two sequential gradient computations during each training step; as mentioned earlier, there are methods for reducing SAM computational overhead that could be integrated in future work.

3) Flexibility to Length: Lastly, we explore the capability of our model to process longer audio during inference. We start by evaluating the audio-text retrieval task on the Clotho dataset, where audio sample duration ranges from 15 to 30 seconds. We compare the performance between (a), which uses the entire length of the audio, and (e), which randomly samples 10-second spectrogram MAE patches from the full audio to match the training duration. We find that using longer sequence lengths leads to an improvement in retrieval recall, even though our training only uses 10-second time-frequency patches. Then, we extend our investigation to include long sequence classification tasks from the HEAR benchmark, specifically focusing on the GTZAN [63] and VoxLingua [64] tasks, as shown in Table XII. Our findings reveal that leveraging the full length of the audio samples consistently yields higher classification accuracy compared to using fixed length patches, which is consistent with findings in FLIP [23].

VII. CONCLUSIONS

In this paper, we presented dataset and model improvements for contrastive audio-language models. By automatically captioning unlabeled audio data and refining noisy existing captions with language models, we curated a dataset of 3.9 million text-audio pairs. Our modeling approach consists of a two-stage training strategy: The first stage involves training an audio encoder with an MAE objective, while the second stage combines contrastive learning with an auxiliary captioning task. Our final model, Cacophony, achieves state-of-the-art performance on audio-text retrieval tasks and competitive performance on other downstream tasks.

We conducted a series of ablation studies to evaluate the effects of our dataset curation and modeling strategies. Notably, we observed positive effects of our modeling techniques and dataset scaling in both training phases. The improvement seen from scaling MAE pretraining data is particularly relevant, as MAE training leverages audio-only data and can easily scale further in future work.

While our work demonstrates significant advancements in audio-text modeling, it is important to acknowledge several limitations. We observe weaker performance in fine-grained classification and captioning tasks, suggesting the need for more detailed, higher-quality captions in our training data. In particular, the reliance on a single-modality audio captioner for synthesizing a large portion of our data may limit the diversity and richness of our dataset. Integrating multimodal captioning synthesis techniques, such as those involving the visual modality, could potentially address this limitation. Furthermore, our model's relatively weak performance on speech-related tasks indicates a need for more comprehensive speech data integration. Finally, the persistent modality gap between audio and text embeddings, for our proposed model and other audio-text models, remains a challenge, potentially limiting performance in tasks requiring tight cross-modal alignment.

REFERENCES

[1] W. Wang, *Machine Audition: Principles, Algorithms and Systems: Principles, Algorithms and Systems*. Hershey, PA, USA: IGI Global, 2010.
[2] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, 2020.
[3] J. Gardner, S. Durand, D. Stoller, and R. Bittner, "LLark: A Multimodal Instruction-Following Language Model for Music," in *Proc. Int. Conf. Mach. Learn.*, 2024.
[4] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. R. Glass, "Listen, think, and understand," in *Proc. Int. Conf. Learn. Representations*, 2024.
[5] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An audio language model for audio tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 18090–18108.
[6] Y. Chu et al., "Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models," 2023, *arXiv:2311.07919*.
[7] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 28492–28518.
[8] R. Huang et al., "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 13916–13932.
[9] H. Liu et al., "AudioLDM: Text-to-audio generation with latent diffusion models," in *Proc. 40th Int. Conf. Mach. Learn.*, Jul. 2023, pp. 21450–21474.
[10] X. Liu et al., "Separate anything you describe," 2023, *arXiv:2308.05037*.

[11] H.-W. Dong, N. Takahashi, Y. Mitsufuji, J. McAuley, and T. Berg-Kirkpatrick, "CLIPSep: Learning text-queried sound separation with noisy unlabeled videos," in *Proc. Int. Conf. Learn. Representations*, 2023.
[12] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics:Human Lang. Technol.*, 2019, pp. 119–132.
[13] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 736–740.
[14] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
[15] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 11975–11986.
[16] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. Ellis, "MuLan: A joint embedding of music audio and natural language," in *Proc. 23rd Int. Soc. Music Inf. Retriev. Conf.*, Bengaluru, India, 2022, pp. 559–566.
[17] 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," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[18] J. F. Gemmeke et al., "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 776–780.
[19] X. Mei et al., "WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 32, pp. 3339–3354, 2024.
[20] 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 *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 646–650.
[21] Y. Liu et al., " RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[22] C.-F. Yeh, P.-Y. Huang, V. Sharma, S.-W. Li, and G. Gosh, "FLAP: Fast language-audio pre-training," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2023, pp. 1–8.
[23] Y. Li, H. Fan, R. Hu, C. Feichtenhofer, and K. He, "Scaling language-image pre-training via masking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23390–23400.
[24] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.
[25] P.-Y. Huang et al., "Masked autoencoders that listen," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 28708–28720.
[26] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, "CoCa: Contrastive captioners are image-text foundation models," *Trans. Mach. Learn. Res.*, 2022.
[27] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 12888–12900.
[28] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do CIFAR-10 classifiers generalize to CIFAR-10?," 2018, *arXiv:1806.00451*.
[29] S. Lipping, P. Sudarsanam, K. Drossos, and T. Virtanen, "Clotho-AQA: A crowdsourced dataset for audio question answering," in *Proc. Eur. Signal Process. Conf.*, 2022, pp. 1140–1144.
[30] J. Turian et al., "HEAR: Holistic evaluation of audio representations," in *Proc. Int. Conf. Neural Inf. Process. Syst. Competitions Demonstrations Track*, 2022, pp. 125–145.
[31] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Trans. Multimedia*, vol. 25, pp. 2675–2685, 2022.
[32] S. Deshmukh, B. Elizalde, and H. Wang, "Audio retrieval with WavText5K and CLAP training," in *Proc. INTERSPEECH*, 2022, pp. 2948–2952.
[33] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.
[34] S. Lee et al., "ACAV100 M: Automatic curation of large-scale datasets for audio-visual video representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10274–10284.
[35] X. Xu et al., "BLAT: Bootstrapping language-audio pre-training based on audioset tag-guided synthetic data," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 2756–2764.
[36] M. Singh et al., "The effectiveness of MAE pre-pretraining for billion-scale pretraining," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 5484–5494.
[37] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[38] X. Li, W. Wang, L. Yang, and J. Yang, "Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality," 2022, *arXiv:2205.10063*.
[39] C. Jia et al., "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4904–4916.
[40] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[41] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3744–3753.
[42] K. Koutini, J. Schlüter, H. Eghbal-Zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Proc. INTERSPEECH,* 2022, pp. 2753–2757.
[43] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019.
[44] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022, *arXiv:2204.06125*.
[45] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *Proc. Int. Conf. Learn. Representations*, 2021.
[46] Y. Liu, S. Mai, X. Chen, C.-J. Hsieh, and Y. You, "Towards efficient and scalable sharpness-aware minimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12360–12370.
[47] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. 23rd Annu. ACM Conf. Multimedia*, pp. 1015–1018.
[48] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.
[49] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50 k: An open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 829–852, 2021.
[50] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "VGGSound: A large-scale audio-visual dataset," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, 2020, pp. 721–725.
[51] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "CLAP:Learning audio concepts from natural language supervision," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, 2023, pp. 1–5.
[52] G. Li, Y. Xu, and D. Hu, "Multi-scale attention for audio question answering," in *Proc. INTERSPEECH*, 2023, pp. 3442–3446.
[53] G. Li, Y. Wei, Y. Tian, C. Xu, J.-R. Wen, and D. Hu, "Learning to answer questions in dynamic audio-visual scenarios," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19108–19118.
[54] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. Detection Classification Acoustic Scenes Events Workshop*, 2018, Art. no. 9.
[55] A. L. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, 2019, pp. 3852–3856.
[56] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12449–12460.
[57] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 161–165.
[58] C. Hawthorne et al., "Onsets and frames: Dual-objective piano transcription," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 50–57.
[59] X. Chen et al., "Microsoft COCO captions: Data collection and evaluation server," 2015, *arXiv:1504.00325*.
[60] V. W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Y. Zou, "Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 17612–17625.
[61] M. Luong, K. Nguyen, N. Ho, R. Haf, D. Phung, and Lizhen Qu, "Revisiting deep audio-text retrieval through the lens of transportation," in *Int. Conf. Learn. Representations*, 2024.
[62] M. Tschannen, M. Kumar, A. P. Steiner, X. Zhai, N. Houlsby, and L. Beyer, "Image captioners are scalable vision learners too," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, 2023, pp. 46830–46855.
[63] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.
[64] J. Valk and T. Alumäe, "VoxLingua107: A dataset for spoken language recognition," in *Proc. 2021 IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 652–658.