Audio-Visual Generalized Zero-Shot Learning using Pre-Trained Large Multi-Modal Models

David Kurzendörfer[*1,2], Otniel-Bogdan Mercea[*1,3,4], A. Sophia Koepke[1,3], Zeynep Akata[3,4,5]

[1]University of Tübingen [2]Localyzer GmbH [3]Tübingen AI Center [4]Helmholtz Munich
[5]Technical University of Munich

{otniel-bogdan.mercea, a-sophia.koepke}@uni-tuebingen.de

dk@localyzer.de, zeynep.akata@helmholtz-munich.de

Abstract

Audio-visual zero-shot learning methods commonly build on features extracted from pre-trained models, e.g. video or audio classification models. However, existing benchmarks predate the popularization of large multi-modal models, such as CLIP and CLAP. In this work, we explore such large pre-trained models to obtain features, i.e. CLIP for visual features, and CLAP for audio features. Furthermore, the CLIP and CLAP text encoders provide class label embeddings which are combined to boost the performance of the system. We propose a simple yet effective model that only relies on feed-forward neural networks, exploiting the strong generalization capabilities of the new audio, visual and textual features. Our framework achieves state-of-the-art performance on VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} with our new features. Code and data available at: https://github.com/dkurzend/ClipClap-GZSL.

1. Introduction

The synergy of audio and visual modalities is a valuable asset for tasks like video classification. Imagine a bustling street captured on camera, where the integration of audio—such as footsteps, car engines, or a dog barking—provides crucial context for interpreting the visual content. In practical deep learning applications, models often encounter new and unseen data, e.g. objects or scenes not present in their training data. This challenge arises due to the vast diversity of real-world data and the impracticality of preparing models for every possible variation. A well-designed deep learning model should exhibit the ability to transfer knowledge
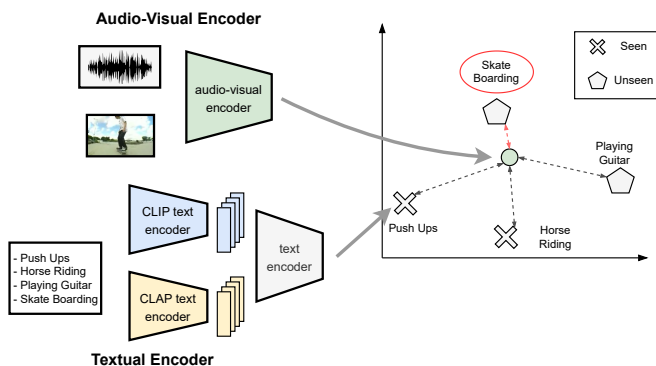
---

[*]Denotes equal contribution

Figure 1. Our framework for audio-visual GZSL maps the audio and visual data to embeddings that are aligned with class label embeddings that are obtained from merging CLIP and CLAP embeddings. The class label embedding that is closest to the audio-visual embedding determines the class prediction. At test time, the set of class label embeddings contains both seen and unseen classes.

from familiar classes to unseen ones.

Audio-visual generalized zero-shot learning (GZSL) aims at classifying videos using audio and visual inputs. Previous work [34, 50, 52, 53, 66] learns to align audio-visual embeddings with corresponding class label embeddings. The class label embedding that is closest to the audio-visual embedding is chosen for the prediction. Existing audio-visual GZSL methods build on features obtained from pre-trained models for the audio, visual and textual data. However, the feature extraction methods [10, 12, 33, 42, 76] used in most previous works [34, 50, 52, 66] do not reflect the state of the art anymore. In recent years, the transformer architecture [78] has proved successful in many areas such as natural language processing [1, 36, 70], the vision domain [21, 70] or the audio domain [16, 28]. CLIP [70] is a popular

vision-language model which contains transformers as the text and image encoders that map to a joint multi-modal embedding space. [51] introduces CLAP, a similar method for the audio-language domain.

In this work, we address audio-visual GZSL by exploring pre-trained multi-modal models to produce audio, visual, and textual input features. We show that the high generalization capabilities of such large pre-trained models are beneficial in the GZSL setting. We use CLIP [70] for visual feature extraction and CLAP [51] for audio feature extraction. Both models contain text encoders which provide input class label embeddings. Consequently, a novel feature of our method compared to prior work (e.g. [34, 50, 52, 53, 66]) is the usage of two class label embeddings that are aggregated into a unified label embedding. Since the textual embeddings are obtained from vision-langauge and audio-language models, the audio and visual input features are already aligned with the corresponding class embeddings. Our proposed model (see Figure 1 for an overview) ingests the aforementioned input features and class label embeddings, only relying on simple feed-forward neural networks in conjunction with a composite loss function. Our contributions can be summarized as follows:

- Our proposed audio-visual GZSL framework builds on features from pre-trained multi-modal models. Moreover, we exploit the text encoders in the same multi-modal models to provide two class label embeddings that are combined to form a unified and robust textual class label embedding;

- Our simple but effective framework achieves state-of-the-art results on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets when using the new input features;

- Qualitative analysis shows that our approach produces well-separated clusters for the seen and unseen classes in the embedding space.

2. Related Work

In this section, we summarize related work concerned with audio-visual learning, zero-shot learning, and audio-visual generalized zero-shot learning.

Audio-Visual Learning. Using audio data for video analysis can significantly enhance the visual representations, for instance for sound source localization [5, 9, 15, 69, 75, 86], sound source separation [26, 77, 93] or both sound source localisation and separation [4, 63, 89, 90]. Various works perform the audio-visual correspondence task [5, 9, 11, 15, 64, 65, 69] or synchronization task [14, 18, 19, 22, 37, 41, 41, 63, 85, 89] to learn representations that contain useful knowledge of both

modalities. Moreover, [8, 10, 17, 18, 54, 58, 67, 85] learn rich audio-visual representations. For example, [8, 10, 18, 67, 85] use self-supervision to learn these rich audio-visual representations, while [17] uses knowledge distillation, and [54, 58] use a supervised learning objective along with a transformer specifically designed to merge the audio and the visual modalities. Multiple works combine the audio and visual modalities for speech recognition and lip reading [2, 3, 48, 57]. Other tasks where audio and visual modalities are combined include spotting of spoken keywords [56] and audio synthesis from visual information [25, 27, 39, 40, 60, 73, 74, 91].

Zero-Shot Learning (ZSL) involves training a model to classify new test classes not seen during training, e.g. by learning a mapping between input features and semantic embeddings. Typically, the semantic embeddings are obtained as text embeddings from class labels [6, 7, 24, 61, 82, 84] and from class attributes and descriptions [43, 71, 84]. Other works use generative methods to synthesize data for unseen classes [29, 30, 32, 79, 84]. In [23, 44, 51, 68, 70, 88, 94], ZSL is performed by applying a pre-trained model on new, unseen datasets. Recently, CLIP text and image encoders have been integrated into various ZSL frameworks [20, 31, 45–47, 62, 80, 81, 87, 92], e.g. for zero-shot / open-vocabulary semantic segmentation [20, 46, 47, 87, 92]. Taking advantage of the strong generalization ability of large pretrained multi-modal models, we use CLIP [70] and CLAP [51] as feature extraction methods for the visual and audio domain.

Audio-visual GZSL was first introduced in [66] which proposed the AudioSetZSL datset. [50, 66] both proposed methods that map the audio, visual, and textual input features (i.e. word2vec [55]) to a joint embedding space. [53] curated several new benchmarks for the audio-visual GZSL task, along with a framework that uses cross-attention between the audio and the visual modalities. [34] additionally uses a hyperbolic alignment loss. While [34, 50, 53, 66] ingest temporally averaged audio and visual features from pre-trained audio and visual classifiers, [52] exploits the inherent temporal structure of videos. In contrast to prior work that fuses the audio and visual information at later stages, our method directly concatenates audio and visual input features before passing them into a feed-forward neural network. Furthermore, our proposed method utilizes two input class label embeddings obtained from CLIP and CLAP.

3. Proposed Approach

In this section, we motivate the use of features extracted from large pre-trained multi-modal models, describe the audio-visual GZSL setting and our proposed framework and training objective.

The audio-visual GZSL benchmarks introduced in prior work [52, 53] build on features extracted from audio and video classification networks. However, those feature extraction methods date back to 2017 and 2015 for the audio [33] and visual features [76] respectively. CLIP [70] and CLAP [51] have shown impressive generalization capabilities. We propose to use features extracted from CLIP and CLAP as inputs to our framework, eliminating the need for a complex architecture to adapt to the audio-visual GZSL task. We use text embeddings obtained from CLIP and CLAP which are aligned with corresponding audio / visual features.

Audio-visual GZSL setting. In the ZSL setting, two disjoint sets of classes are considered, i.e. seen and unseen classes S and U with $S \cap U = \varnothing$. In ZSL, the model is trained on the seen classes and later evaluated on the test set, which only consists of unseen classes. In the GZSL setting the model is trained on seen classes (S), but the test set contains both seen and unseen classes, making this scenario more realistic.

Formally, the set of data samples that belong to the seen classes is denoted by $S = \left(v_i^s, a_i^s, w_i^s, y_i^s\right)_{i \in \{1, \cdots, N\}}$ where each data point i is a quadruple where a_i^s is the audio feature, v_i^s is the visual feature, y_i^s is the ground-truth class label of sample i and w_i^s is the textual label embedding corresponding to the ground-truth class label. N is the number of samples in S. Likewise, the set of samples from unseen classes of size M is defined as $U = \left(v_i^u, a_i^u, w_i^u, y_i^u\right)_{i \in \{1, \cdots, M\}}$. In GZSL, the goal is to learn a function $h : \left(v_i^s, a_i^s\right) \mapsto w_i^s$ which for samples from unseen classes fulfills $h\left(v_i^u, a_i^u\right) = w_i^u$. The total number of classes is denoted as K and the class label $j \in \{1, 2 \ldots, K\}$. The number of seen and unseen classes are denoted as K_s and K_u.

Model architecture. Our proposed model is visualized in Fig. 2. It accepts audio and visual features as inputs, denoted as $a \in \mathbb{R}^{d_{in_a}}$ and $v \in \mathbb{R}^{d_{in_v}}$ respectively (for simplicity, subscripts i denoting the ith sample are dropped). These features are obtained using CLAP and CLIP as feature extractors. In addition, our model takes as input text embeddings $w^v \in \mathbb{R}^{d_{in_v}}$ from CLIP and $w^a \in \mathbb{R}^{d_{in_a}}$ from CLAP. CLIP and CLAP merely serve as feature extractors and thus are not optimized when training our framework. Our proposed model consists of a branch for the audio-visual features, and a branch for the textual label embeddings. In the audio-
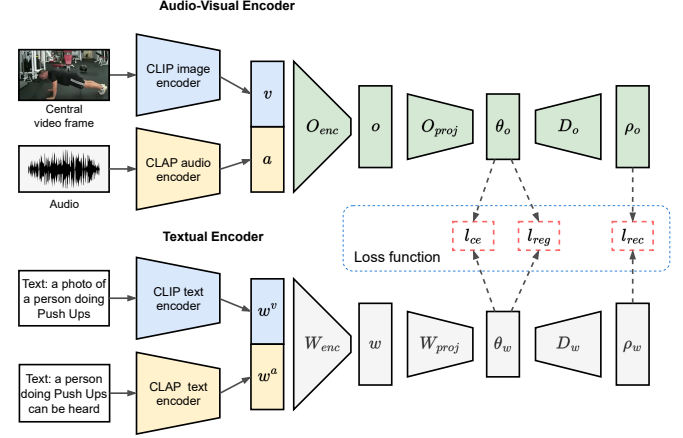

Figure 2. The image and audio encoders of CLIP and CLAP are used to extract features from the raw input which are concatenated and passed through multiple feed-forward networks to get an audio-visual output embedding θ_o. Likewise, the text encoders of CLIP and CLAP are used to extract textual label embeddings. They are passed through a series of neural networks to obtain a learned class label embedding θ_w. Both θ_o and θ_w reside in a joint embedding space.

visual branch, the inputs a and v are first concatenated and then passed through an encoder block O_{enc} to produce the audio-visual input

$$o = O_{enc}(\text{concat}(v, a)), \qquad (1)$$

where $o \in \mathbb{R}^{d_{model}}$. O_{enc} consists of a linear layer $f_{O_{enc}} : \mathbb{R}^{(d_{in_v} + d_{in_a})} \to \mathbb{R}^{d_{model}}$, followed by batch normalization [35], a ReLU activation function [59], and dropout [72]. To get the final audio-visual embedding $\theta_o \in \mathbb{R}^{d_{out}}$ that is used for the prediction, o is passed through a projection network $\theta_o = O_{proj}(o)$, where θ_o is composed of linear layers $f_{O_{proj}}^1 : \mathbb{R}^{d_{model}} \to \mathbb{R}^{d_{hidden}}$ and $f_{O_{proj}}^2 : \mathbb{R}^{d_{hidden}} \to \mathbb{R}^{d_{out}}$. Both layers are followed by batch normalization, ReLU, and dropout.

The textual branch follows a similar structure as the audio-visual branch. First, w^a and w^v are concatenated and are input into an encoder network to generate a unified text embedding $w \in \mathbb{R}^{d_{model}}$,

$$w = W_{enc}(\text{concat}(w^v, w^a)). \qquad (2)$$

W_{enc} contains a linear layer $f_{W_{enc}} : \mathbb{R}^{(d_{in_v} + d_{in_a})} \to \mathbb{R}^{d_{model}}$ followed by batch normalization, ReLU, and dropout. The output w is further processed by a projection layer $\theta_w = W_{proj}(w)$, where $\theta_w \in \mathbb{R}^{d_{out}}$. W_{proj} is given by a linear layer $f_{W_{proj}} : \mathbb{R}^{d_{model}} \to \mathbb{R}^{d_{out}}$ with batch normalization, ReLU, and dropout. The goal of the model is to align the projected label embedding θ_w and the audio-visual output embedding θ_o in a joint embedding space of dimension d_{out}, such that θ_o is closest to the θ_w that corresponds to the ground-truth class.

At test time, classification is done by calculating θ_w for all the classes and determining the class label c that is closest to θ_o:

$$c = \underset{j}{\arg\min} \left(\left\| \theta_w^j - \theta_o \right\|_2 \right), \tag{3}$$

where θ_w^j is the output label embedding θ_w for class j.
Training objective. The loss function l used to train our framework is adopted from [52] and consists of a cross-entropy loss l_{ce}, a reconstruction loss l_{rec}, and a regression loss l_{reg}. The final loss is then given by

$$l = l_{ce} + l_{rec} + l_{reg}. \tag{4}$$

The **cross-entropy loss** is given by

$$l_{ce} = -\frac{1}{n} \sum_{i}^{n} \log \left(\frac{\exp\left(\theta_{w_{seen},k_{gt_i}} \theta_{o_i}\right)}{\sum_{k_j}^{K_s} \exp\left(\theta_{w_{seen},k_j} \theta_{o_i}\right)} \right), \tag{5}$$

where $\theta_{w_{seen}} \in \mathbb{R}^{K_s \times d_{out}}$ denotes the matrix of the projected class embeddings for the seen classes. $k_{gt} \in \{1, ,2 \ldots, K_s\}$ refers to the ground-truth class index, and thus $\theta_{w_{seen},k_{gt_i}}$ selects the row of $\theta_{w_{seen}}$ that belongs to the target of the current sample i. The number of training samples is denoted by n.

The **reconstruction loss** semantically aligns the output embeddings θ_o and θ_w. For this, two decoder networks, D_o and D_w, are used to obtain $\rho_o = D_o(\theta_o)$ with $\rho_o \in \mathbb{R}^{d_{model}}$. D_o consists of two linear layers $f_{D_o}^1 : \mathbb{R}^{d_{out}} \to \mathbb{R}^{d_{hidden}}$ and $f_{D_o}^2 : \mathbb{R}^{d_{hidden}} \to \mathbb{R}^{d_{model}}$ which are both followed by batch normalization, ReLU, and dropout. D_w gets θ_w as input, such that $\rho_w = D_w(\theta_w)$, where $\rho_w \in \mathbb{R}^{d_{model}}$. D_w consists of one linear layer $f_{D_w} : \mathbb{R}^{d_{out}} \to \mathbb{R}^{d_{model}}$ with batch normalization, ReLU, and dropout.

The reconstruction loss encourages the reconstructions ρ_o and ρ_w to be close to the label embedding w by minimizing

$$l_{rec} = \frac{1}{n} \sum_{i=1}^{n} (\rho_{o_i} - w_i)^2 + \frac{1}{n} \sum_{i=1}^{n} (\rho_{w_i} - w_i)^2, \tag{6}$$

where n is the number of training samples.

The **regression loss** computes the mean squared error between the output embeddings of the model and the ground-truth label embeddings:

$$l_{reg} = \frac{1}{n} \sum_{i=1}^{n} (\theta_{o_i} - \theta_{w_i})^2, \tag{7}$$

where θ_{o_i} is the audio-visual embedding for sample i and θ_{w_i} is the corresponding output label embedding.

4. Experiments

In this section, we describe our experimental setup (Sec. 4.1), our results (Sec. 4.2), and ablate crucial components of our framework (Sec. 4.3).

4.1. Experimental setup

Here, we describe the evaluation metrics, the features used, implementation details, and our baselines.

Evaluation metrics. We evaluate our audio-visual GZSL method on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets introduced in [53] and as suggested in [52] (instead of using the *main split* also introduced in [53]). We follow [52, 53, 83] and report the mean class accuracy scores for the seen classes (acc_S) and unseen classes (acc_U) separately. For the GZSL performance metric, their harmonic mean is obtained as

$$HM = \frac{2 * acc_U * acc_S}{acc_U + acc_S}. \tag{8}$$

In addition, we calculate the zero-shot learning performance as the mean class accuracy acc_{ZSL} for the unseen classes. In this setting, only classes from the subset of unseen test classes can be selected as prediction.

Feature extraction. We do not rely on the same feature extractors as previous work [34, 52, 53]. Instead, visual features v_i and class label embeddings w_j^v are extracted from the videos using **CLIP** [70]. For each video, the middle frame is passed through the image encoder of ViT-B/32 CLIP model, yielding a $512-$dimensional feature vector. In addition, for each class label, a $512-$dimensional textual embedding is extracted using the CLIP text encoder. Here, we follow [70], which recommends the usage of text prompt ensembles. We provide more details and a concrete list of text prompts in the supplementary materials in A.1.

Likewise, audio features a_i and class label embeddings w_j^a are extracted using **CLAP** [51]. The raw audio data is resampled to 32000 Hz and center cropped or zero-padded to 10 seconds, depending on the audio length. 64-dimensional log mel-spectrograms are extracted from the audio by using a 1024-point Hanning window with a hop size of 320. Audio embeddings are obtained from the audio encoder of CLAP, and text embeddings from its ext encoder. The joint audio and textual embedding space in CLAP is of size 1024. For the class text embeddings, text prompt ensembles are used similar to CLIP (See more details in the supplementary materials A.2).

Implementation details. To train our framework, we use the Adam optimizer [38] with weight decay $1e^{-5}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a batch size of

64. Furthermore, the initial learning rates are $1e^{-4}$ / $7e^{-5}$ / $1e^{-4}$ for VGGSound-GZSL^{cls} / UCF-GZSL^{cls} / and ActivityNet-GZSL^{cls}. When the validation HM score does not improve for 3 consecutive epochs during training, the learning rate is reduced by a factor of 0.1. In the first training stage, the models for VGGSound-GZSL^{cls} and ActivityNet-GZSL^{cls} were trained for 15 epochs, while for UCF-GZSL^{cls} we used 20 epochs. Calibrated stacking [13] is a scalar which biases the output of the network towards unseen classes, as the network without the calibration is significantly biased towards seen classes. To address the inherent bias of ZSL methods towards seen classes, we use calibrated stacking with search space interval [0, 5] and a step size of 0.07.

We follow the training and evaluation protocol of [34, 52, 53] for our method. The training is divided into two stages. In the first stage, models are trained on the training set. The validation set comprising val(U) and val(S) is used to determine model parameters such as those for calibrated stacking and the best epoch based on the HM score. In the second training stage, the models are trained again using the parameters determined in the first stage, however, this time, the models are trained on the union of the training and validation set {train ∪ val(S) ∪ val(U)}. Finally, the final results on the test set are obtained by evaluating the models trained in the second stage.

The inputs are of size d_{in_a} = 1024 and d_{in_v} = 512. For the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets, the model dimension d_{model} = 512 is chosen, and the output dimension is set to d_{out} = 64. In addition, for all three datasets, d_{hidden} is set to 512, and the dropout rate is set to 0.1. All models were trained on a single NVIDIA GeForce RTX 2080 Ti GPU.

Baselines. We compare our framework with the state-of-the-art methods CJME [66], AVGZSLNet [50], AVCA [53], and Hyper-multiple [34]. For CJME, AVGZSLNet and AVCA we use the training parameters from [53] and we evaluate Hyper-multiple using the training parameters from [34]. All methods are evaluated using the new CLIP and CLAP features. For fairness, we adjust the baseline methods for our input representation to using two textual input embeddings, by appending an additional layer at the beginning of the network as

$$w_i = W_{enc}(\text{concat}\left(w_i^v, w_i^a\right)), \qquad (9)$$

where $w_i^v \in \mathbb{R}^{512}$ is the CLIP class label embedding for sample i and $w_i^a \in \mathbb{R}^{1024}$ is the CLAP class label embedding.

4.2. Experimental results

In this section, we present quantitative and qualitative results for audio-visual GZSL obtained with our proposed framework.

Quantitative results. On all three datasets, our model outperforms all baseline methods in terms of GZSL performance (HM) as can be seen in Tab. 1. For example, on UCF-GZSL^{cls}, we achieve a HM score of 55.97% whereas the next best baseline (AVGZSLNet) achieves 42.67%. Similarly, on ActivityNet-GZSL^{cls}, our model achieves a HM score of 27.93% while Hyer-multiple achieves 20.90%, showing an improvement of 7.03%. Finally, on VGGSound-GZSL^{cls}, our model achieves 16.18%, compared to 11.87% for Hyper-multiple.

Our method also outperforms all the baselines on all three datasets for ZSL (acc_{ZSL}). On VGGSound-GZSL^{cls}, we achieve a acc_{ZSL} score of 11.53% while the second-best method achieves a score of 8.47%. On UCF-GZSL^{cls}, our method achieves a acc_{ZSL} performance of 46.96%, compared to 40.28% for Hyper-multiple. On ActivityNet-GZSL^{cls}, the difference in ZSL performance is very small. Our model obtains a acc_{ZSL} score of 22.76% while Hyper-multiple achieves 22.18%.

In terms of the seen and unseen scores acc_S and acc_U, our method is the best performing model most of the time. On VGGSound-GZSL^{cls} we achieve the second best acc_S of 29.68%, while AVCA achieves 32.47%. For the acc_U, our model performs best with 11.12% vs. 8.12% achieved by Hyper-multiple. On UCF-GZSL^{cls}, our model achieves the highest acc_S / acc_U scores with 77.14% / 43.91% compared to 56.26% / 39.77% achieved by AVGZSLNet / Hyper-multiple. On ActivityNet-GZSL^{cls}, we achieve the best acc_S score with 45.98%, compared to 24.04% achieved by AVCA. For the acc_U performance, only Hyper-multiple performs better than our method with 21.30% vs. 20.06%.

The GZSL and ZSL results show, that when using CLIP and CLAP as feature extraction methods, a rather simple model like our method, is able to outperform methods that use more sophisticated architectural components such as cross-attention used in AVCA and Hyper-multiple, or complex concepts from hyperbolic geometry used in Hyper-multiple.

Our method uses around 2.2 million parameters, whereas CJME and AVGZSLNet use 2.3 million parameters, and AVCA and Hyper-multiple have approximately 2.4 million parameters. These numbers do not include the number of parameters of the feature extraction methods CLIP and CLAP.

Overall, our model achieves significant improvements over the baselines, while it requires marginally

Method	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}
CJME [66]	11.96	5.41	7.45	6.84	48.18	17.68	25.87	20.46	16.06	9.13	11.64	9.92
AVGZSLNet [50]	13.02	2.88	4.71	5.44	56.26	34.37	42.67	35.66	14.81	11.11	12.70	12.39
AVCA [53]	**32.47**	6.81	11.26	8.16	34.90	38.67	36.69	38.67	24.04	19.88	21.76	20.88
Hyper-multiple [34]	21.99	8.12	11.87	8.47	43.52	39.77	41.56	40.28	20.52	**21.30**	20.90	22.18
Ours	29.68	**11.12**	**16.18**	**11.53**	**77.14**	**43.91**	**55.97**	**46.96**	**45.98**	20.06	**27.93**	**22.76**

Table 1. Performance of our model compared to state-of-the-art methods for audio-visual (G)ZSL on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets. For a fair comparison, all baselines are also trained and evaluated using both CLIP and CLAP features and class label embeddings. We report the mean class accuracy for seen (acc_S) and unseen (acc_U) classes, along with their harmonic mean (HM) for GZSL performance. In addition, ZSL performance (acc_{ZSL}) is reported.

fewer parameters. Furthermore, unlike AVCA [53] and Hyper-multiple [34], our method does not require positive and negative samples to calculate a triplet loss during training. This reduces memory requirements and allows for a larger batch size.

Qualitative results. We provide t-SNE visualisations (Fig. 3) of our learned output embeddings on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets. Two unseen classes and four seen classes were randomly selected from the test set. For the unseen classes, all samples were used for the visualization, for the seen classes, all samples from the test set were used. This results in a class imbalance in the plots, since some seen classes have only a few test samples.

It can be observed that while the input features are not clustered well for all datasets, the t-SNE plots for the model outputs shows well-separated clusters for all classes. In particular, the unseen classes are very well-separated. This shows that our method learns useful embeddings for both seen and unseen classes. Only on VGGSound-GZSL^{cls}, our model does not separate well the unseen class *wood thrush calling* and the seen class *barn swallow calling*. This might come from the fact, that both classes can be categorized as bird sounds. Finally, all the text embeddings are located inside the cluster of the class they belong to. This shows that our approach effectively learns to assign the audio-visual input features to the correct class.

4.3. Ablation studies

In this section, we conduct ablations studies on different model design choices. First, we ablate the choice of using two different textual label embeddings. Then, we study the benefit of using multi-modal inputs (audio and visual). Thirdly, we analyze the influence of the different loss components.

Class label embeddings. Our proposed framework uses two class label embeddings as inputs. Here, we compare this strategy to the usage of only a single class

label embedding. For this, we train our model only using CLIP text embeddings w^v, or only using CLAP text embeddings w^a. However, we use both audio and visual input features for this experiment. The results are presented in Tab. 2.

Using either only w^v or only w^a as input class label embedding leads to very similar results on VGGSound-GZSL^{cls} and UCF-GZSL^{cls}. However, our proposed method that uses both w^v and w^a, outperforms them significantly. On UCF-GZSL^{cls}, using both text embeddings obtains a HM 55.97% vs. 45.83% for w^v, while on VGGSound^{cls} our method obtains a HM of 16.18% vs. 13.37% for w^v. The same trends can be observed for the ZSL performance. On ActivityNet-GZSL^{cls}, using w^v leads to HM / acc_{ZSL} scores of 28.45% / 23.18%, while using both w^v and w^a performs slightly worse, achieving HM / acc_{ZSL} scores of 27.93% / 22.76%.

Finally, using both label embeddings help significantly in terms of the acc_U score. For VGGSound-GZSL^{cls}, we boost performance from 8.94% for w^a to 11.12%, while on UCF-GZSL^{cls} we improve the performance from 39.62% for w^a to 43.91% when using both label embeddings (Both). On ActivityNet-GZSL^{cls}, using both embeddings gives slightly lower numbers than using only w^v in terms of the acc_U scores. On the other hand, our method obtains the best acc_S results on all three datasets. Overall, jointly using both w^a and w^v provides a significant boost in performance across all the metrics and datasets.

Multi-modality. In Tab. 3, we present the impact of using multi-modal input data. To obtain results for using a single input modality, only the audio or visual input feature (a or v) along with the corresponding text embedding (w^a or w^v) is used.

On VGGSound-GZSL^{cls}, using only the audio modality achieves higher HM and acc_{ZSL} scores compared to using the visual modality with HM / acc_{ZSL} scores of 11.99% / 9.34% vs. 9.62% / 7.16% for the visual modality. This is likely due to VGGSound being curated specifically to include relevant audio informa-
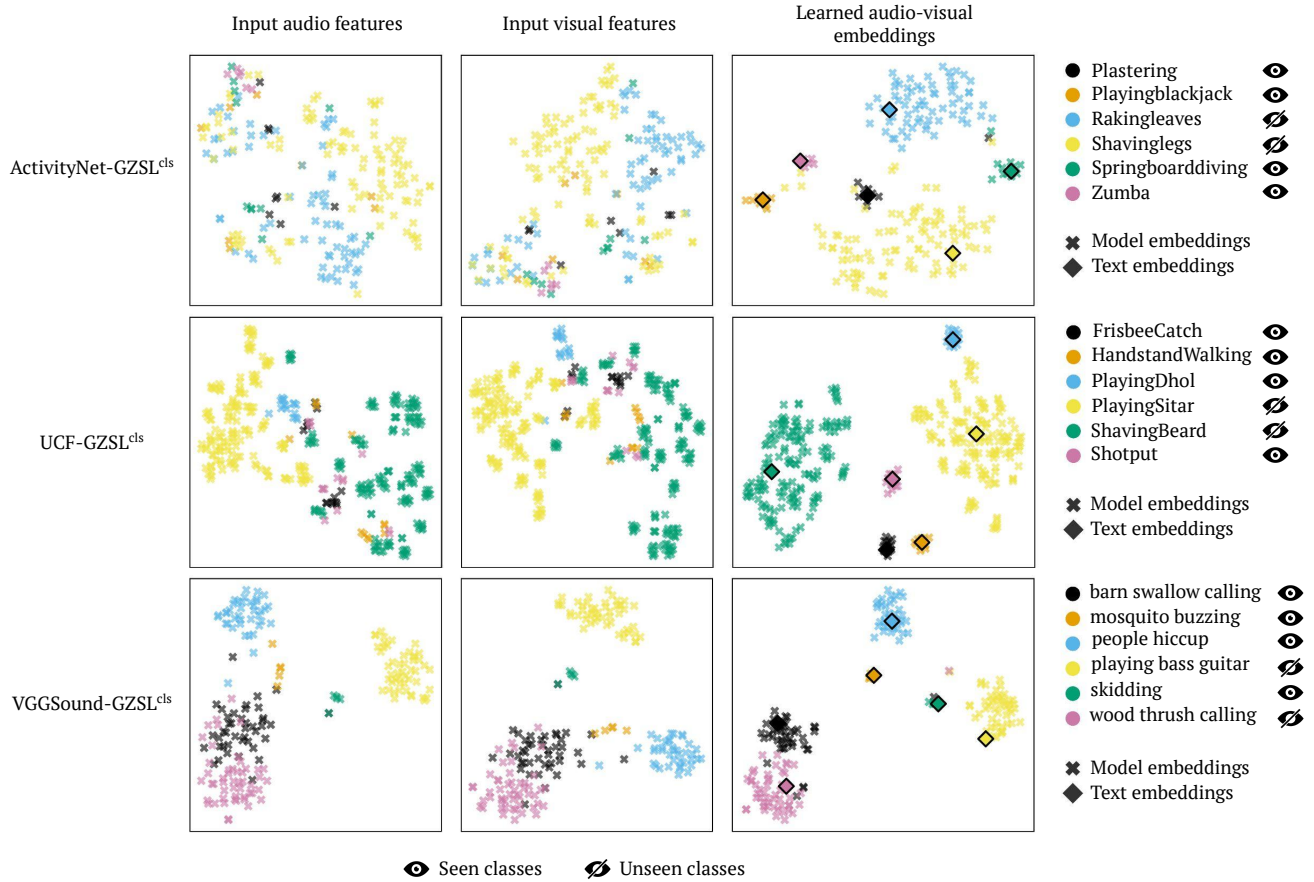
Figure 3. t-SNE visualizations for the audio input features (left), visual input features (center), and the learned output embeddings for our model (right) for the ActivityNet-GZSL^{cls} (top), UCF-GZSL^{cls} (center) and VGGSound-GZSL^{cls} (bottom) datasets for two unseen classes and four seen classes. The learned class text embeddings are visualized as diamonds.

Label Embedding	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}
CLIP (w^v)	28.30	8.75	13.37	9.28	75.91	32.83	45.83	37.47	43.91	**21.04**	**28.45**	**23.18**
CLAP (w^a)	18.71	8.94	12.10	9.09	53.09	39.62	45.38	39.78	35.08	13.03	19.00	14.20
Both (Ours)	**29.68**	**11.12**	**16.18**	**11.53**	**77.14**	**43.91**	**55.97**	**46.96**	**45.98**	20.06	27.93	22.76

Table 2. Influence of using the two input label embeddings from CLIP and CLAP on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets.

tion. In contrast, for UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}, using only the visual modality achieves better results than the audio modality on its own. On ActivityNet-GZSL^{cls}, the audio modality results in HM / acc_{ZSL} scores of 8.15% / 6.75% while using visual inputs gives HM / acc_{ZSL} scores of 26.69% / 22.58%.

Across all datasets, the acc_S score is significantly improved when using both modalities compared to using only a or v. On UCF-GZSL^{cls}, our full model (Both) yields a acc_S performance of 77.14% vs. 53.65% for v and 35.59% for a. For the acc_U score, we slightly improve upon the v, and significantly improve over

a. The same trend can be observed on VGGSound^{cls} where our model (Both) significantly outperforms both a and v in acc_U and acc_S. On ActivityNet^{cls}, our full model is significantly stronger in terms of the acc_S score, while it is slightly outperformed in terms of acc_U when using only the visual modality v.

Overall, using both modalities as inputs is sound and leads to the best performance. These results highlight the fact that our model effectively exploits cross-modal relationships through the fusion of audio and visual modalities by using linear layers.

Training objective. We present results for using

Modality	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}
Audio (a)	17.48	9.12	11.99	9.34	35.59	39.69	37.53	41.13	10.72	6.58	8.15	6.75
Visual (v)	15.39	7.00	9.62	7.16	53.65	43.13	47.82	43.98	38.59	**20.40**	26.69	22.58
Both (Ours)	**29.68**	**11.12**	**16.18**	**11.53**	**77.14**	**43.91**	**55.97**	**46.96**	**45.98**	20.06	**27.93**	**22.76**

Table 3. Influence of using only one modality or both modalities as inputs for our method on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets.

Loss	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}
l_{reg}	5.41	9.44	6.87	10.03	22.76	25.49	24.05	28.22	5.01	6.69	5.73	7.11
$l_{reg} + l_{ce}$	**32.64**	**11.91**	**17.45**	**12.47**	76.79	40.30	52.86	43.16	38.93	**20.37**	26.75	22.73
$l_{reg} + l_{ce} + l_{rec}$	29.68	11.12	16.18	11.53	**77.14**	**43.91**	**55.97**	**46.96**	**45.98**	20.06	**27.93**	**22.76**

Table 4. Influence of using different components of the loss function on the (G)ZSL performance for the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets.

different loss functions in Tab. 4. Only using the regression loss l_{reg} yields the poorest performance on all three datasets, with HM scores of 6.87% / 24.05% / 5.73% for VGGSound-GZSL^{cls} / UCF-GZSL^{cls} / ActivityNet-GZSL^{cls}. Using the cross-entropy loss l_{ce} in addition to the regression loss drastically improves the performance with HM scores of 17.45% / 52.86% / 26.75% on VGGSound-GZSL^{cls} / UCF-GZSL^{cls} / ActivityNet-GZSL^{cls}. Finally, adding the reconstruction loss, i.e. when using the full loss function $l_{reg} + l_{ce} + l_{rec}$, we achieve the best overall GZSL results. While the impact of the reconstruction loss is smaller compared to the other two components, it still bring gains in performance. We observe a similar pattern for acc_{ZSL}.

Furthermore, on VGGSound-GZSL^{cls}, acc_S heavily benefits from adding the cross-entropy loss function. For all three datasets, one can observe at least a three-fold improvement. Moreover, we see improvements in the acc_U, where l_{ce} brings significant improvements. On UCF-GZSL^{cls}, our proposed loss function performs best in terms of both acc_S and acc_U. On ActivityNet-GZSL^{cls}, the complete loss function achieves the best acc_S, while the $l_{reg} + l_{ce}$ loss function gives the best acc_U scores. Finally, on VGGSound-GZSL^{cls}, $l_{reg} + l_{ce}$ obtains a slightly better acc_S and acc_U score than our full loss. Overall, this shows that the full training objective provides the best results across all evaluation metrics.

5. Limitations

Our proposed method sets the new state of the art for audio-visual ZSL on three benchmark datasets when using CLIP and CLAP features. However, since the dataset used to train CLIP is not publicly available, we cannot guarantee that no unsseen classes were used. Similarly, we did not attempt to remove unseen classes from the *WavCaps* dataset used to train CLAP. However, [49] shows that information leakage from CLIP pre-training to image ZSL is not very significant. Incorporating CLIP encoders into the model architecture is already an established practice in current research in zero-shot / open-vocabulary semantic segmentation [20, 46, 47, 87, 92]. As CLIP and CLAP were not specifically trained for the task of audio-visual GZSL, our problem setting requires significant transfer of knowledge to the new task.

6. Conclusion

In this paper, we explored the usage of pre-trained large multi-modal models for audio-visual generalized zero-shot learning. Our proposed framework ingests features extracted from the CLIP [70] and CLAP [51] models. One of the advantages of both of the feature extraction methods is that they are also able to produce textual input embeddings for the class labels. We proposed a simple model that consists of feed-forward neural networks and is trained with a composite loss function. When utilizing input features and both label embeddings obtained from CLIP and CLAP, our method achieves state-of-the-art results on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. In *IEEE TPAMI*, 2018. 2

[3] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Asr is all you need: Cross-modal distillation for lip reading. In *ICASSP*, 2020. 2

[4] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *ECCV*, 2020. 2

[5] Triantafyllos Afouras, Yuki M Asano, Francois Fagan, Andrea Vedaldi, and Florian Metze. Self-supervised object detection from audio-visual correspondence. In *CVPR*, 2022. 2

[6] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. In *IEEE TPAMI*, 2015. 2

[7] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015. 2

[8] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020. 2

[9] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *ECCV*, 2018. 2

[10] Yuki Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020. 1, 2

[11] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016. 2

[12] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[13] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. 5

[14] Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. Audio-visual synchronisation in the wild. In *BMVC*, 2021. 2

[15] Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. Localizing visual sounds the hard way. In *CVPR*, 2021. 2

[16] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP*, 2022. 1

[17] Yanbei Chen, Yongqin Xian, A Koepke, Ying Shan, and Zeynep Akata. Distilling audio-visual knowledge by compositional contrastive learning. In *CVPR*, 2021. 2

[18] Ying Cheng, Ruize Wang, Zhihao Pan, Rui Feng, and Yuejie Zhang. Look, listen, and attend: Co-attention network for self-supervised audio-visual representation learning. In *ACM-MM*, 2020. 2

[19] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *ACCV*, 2017. 2

[20] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *CVPR*, 2022. 2, 8

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[22] Joshua P Ebeneze, Yongjun Wu, Hai Wei, Sriram Sethuraman, and Zongyi Liu. Detection of audio-video synchronization errors via event detection. In *ICASSP*, 2021. 2

[23] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP*, 2023. 2

[24] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013. 2

[25] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. In *ECCV*, 2020. 2

[26] Ruohan Gao and Kristen Grauman. Co-separating sounds of visual objects. In *ICCV*, 2019. 2

[27] Shir Goldstein and Yael Moses. Guitar music transcription from silent video. In *BMVC*, 2018. 2

[28] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. In *INTERSPEECH*, 2021. 1

[29] Shreyank N Gowda. Synthetic sample selection for generalized zero-shot learning. In *CVPR*, 2023. 2

[30] Akshita Gupta, Sanath Narayan, Salman Khan, Fahad Shahbaz Khan, Ling Shao, and Joost van de Weijer. Generative multi-label zero-shot learning. In *IEEE TPAMI*, 2023. 2

[31] Lukas Haas, Silas Alberti, and Michal Skreta. Learning generalized zero-shot learners for open-domain image geolocalization. *arXiv preprint arXiv:2302.00275*, 2023. 2

[32] Yun Hao, Yukun Su, Guosheng Lin, Hanjing Su, and Qingyao Wu. Contrastive generative network with recursive-loop for 3d point cloud generalized zero-shot classification. In *Pattern Recognition*, 2023. 2

[33] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore,

Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *ICASSP*, 2017. 1, 3

[34] Jie Hong, Zeeshan Hayder, Junlin Han, Pengfei Fang, Mehrtash Harandi, and Lars Petersson. Hyperbolic audio-visual zero-shot learning. In *ICCV*, 2023. 1, 2, 4, 5, 6

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3

[36] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 1

[37] Naji Khosravan, Shervin Ardeshir, and Rohit Puri. On attention modules for audio-visual synchronization. In *CVPRW*, 2019. 2

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[39] A. Sophia Koepke, Olivia Wiles, and Andrew Zisserman. Visual pitch estimation. In *SMC*, 2019. 2

[40] A. Sophia Koepke, Olivia Wiles, Yael Moses, and Andrew Zisserman. Sight to sound: An end-to-end approach for visual piano transcription. In *ICASSP*, 2020. 2

[41] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 2

[42] Anurag Kumar, Maksim Khadkevich, and Christian Fügen. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 326–330. IEEE, 2018. 1

[43] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. In *IEEE TPAMI*, 2013. 2

[44] Ang Li, Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. Learning visual n-grams from web data. In *ICCV*, 2017. 2

[45] Xiang Li, Congcong Wen, Yuan Hu, and Nan Zhou. Rs-clip: Zero shot remote sensing scene classification via contrastive vision-language supervision. In *Int. J. Appl. Earth Obs. Geoinf.*, 2023. 2

[46] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, 2023. 2, 8

[47] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *ICML*, 2023. 2, 8

[48] Pingchuan Ma, Stavros Petridis, and Maja Pantic. End-to-end audio-visual speech recognition with conformers. In *ICASSP*, 2021. 2

[49] Prasanna Mayilvahanan, Thaddäus Wiedemer, Evgenia Rusak, Matthias Bethge, and Wieland Brendel. Does clip's generalization performance mainly stem from high train-test similarity? In *ICLR*, 2024. 8

[50] Pratik Mazumder, Pravendra Singh, Kranti Kumar Parida, and Vinay P Namboodiri. Avgzslnet: Audio-visual generalized zero-shot learning by reconstructing label features from multi-modal embeddings. In *WACV*, 2021. 1, 2, 5, 6

[51] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *arXiv preprint arXiv:2303.17395*, 2023. 2, 3, 4, 8

[52] Otniel-Bogdan Mercea, Thomas Hummel, A. Sophia Koepke, and Zeynep Akata. Temporal and cross-modal attention for audio-visual zero-shot learning. In *ECCV*, 2022. 1, 2, 3, 4, 5

[53] Otniel-Bogdan Mercea, Lukas Riesch, A. Sophia Koepke, and Zeynep Akata. Audio-visual generalised zero-shot learning with cross-modal attention and language. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6

[54] Otniel-Bogdan Mercea, Thomas Hummel, A. Sophia Koepke, and Zeynep Akata. Text-to-feature diffusion for audio-visual few-shot learning. In *DAGM GCPR*, 2023. 2

[55] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013. 2

[56] Liliane Momeni, Triantafyllos Afouras, Themos Stafylakis, Samuel Albanie, and Andrew Zisserman. Seeing wake words: Audio-visual keyword spotting. In *BMVC*, 2020. 2

[57] Arsha Nagrani, Joon Son Chung, Samuel Albanie, and Andrew Zisserman. Disentangled speech embeddings using cross-modal self-supervision. In *ICASSP*, 2020. 2

[58] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In *NeurIPS*, 2021. 2

[59] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 3

[60] Medhini Narasimhan, Shiry Ginosar, Andrew Owens, Alexei A Efros, and Trevor Darrell. Strumming to the beat: Audio-conditioned contrastive video textures. In *WACV*, 2022. 2

[61] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013. 2

[62] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *ICML*, 2023. 2

[63] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018. 2

[64] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016. 2

[65] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Learning sight from sound: Ambient sound provides supervision for visual learning. In *IJCV*, 2018. 2

[66] Kranti Parida, Neeraj Matiyali, Tanaya Guha, and Gaurav Sharma. Coordinated joint multimodal embeddings for generalized audio-visual zero-shot classification and retrieval of videos. In *WACV*, 2020. 1, 2, 5, 6

[67] Mandela Patrick, Yuki Asano, Polina Kuznetsova, Ruth Fong, Joao F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. In *ICCV*, 2021. 2

[68] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. Combined scaling for zero-shot transfer learning. In *Neurocomputing*, 2023. 2

[69] Rui Qian, Di Hu, Heinrich Dinkel, Mengyue Wu, Ning Xu, and Weiyao Lin. Multiple sound sources localization from coarse to fine. In *ECCV*, 2020. 2

[70] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 3, 4, 8

[71] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 2

[72] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *JMLR*, 2014. 3

[73] Kun Su, Xiulong Liu, and Eli Shlizerman. Multiinstrumentalist net: Unsupervised generation of music from body movements. *arXiv preprint arXiv:2012.03478*, 2020. 2

[74] Kun Su, Xiulong Liu, and Eli Shlizerman. How does it sound? generation of rhythmic soundtracks for human movement videos. 2021. 2

[75] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018. 2

[76] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1, 3

[77] Efthymios Tzinis, Scott Wisdom, Aren Jansen, Shawn Hershey, Tal Remez, Daniel PW Ellis, and John R Hershey. Into the wild with audioscope: Unsupervised audio-visual separation of on-screen sounds. In *ICLR*, 2021. 2

[78] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[79] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, 2018. 2

[80] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *ICCV*, 2023. 2

[81] Meiliu Wu and Qunying Huang. Im2city: image geolocalization via multi-modal learning. In *ACM SIGSPATIAL*, 2022. 2

[82] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 2

[83] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. In *IEEE TPAMI*, 2018. 4

[84] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018. 2

[85] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020. 2

[86] Haoming Xu, Runhao Zeng, Qingyao Wu, Mingkui Tan, and Chuang Gan. Cross-modal relation-aware networks for audio-visual event localization. In *ACM-MM*, 2020. 2

[87] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pretrained vision-language model. In *ECCV*, 2022. 2, 8

[88] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. 2022. 2

[89] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, 2018. 2

[90] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *ICCV*, 2019. 2

[91] Hang Zhou, Ziwei Liu, Xudong Xu, Ping Luo, and Xiaogang Wang. Vision-infused deep audio inpainting. In *ICCV*, 2019. 2

[92] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *CVPR*, 2023. 2, 8

[93] Lingyu Zhu and Esa Rahtu. V-slowfast network for efficient visual sound separation. In *WACV*, 2022. 2

[94] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver:

Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *CVPR*, 2022. 2