Interspeech 2024
1-5 September 2024, Kos, Greece

M2D-CLAP: Masked Modeling Duo Meets CLAP for Learning General-purpose Audio-Language Representation

Daisuke Niizumi[1†], *Daiki Takeuchi*[1], *Yasunori Ohishi*[1], *Noboru Harada*[1], *Masahiro Yasuda*[1], *Shunsuke Tsubaki*[2], *and Keisuke Imoto*[2]

[1] NTT Corporation, [2] Doshisha University, Japan

[†]daisuke.niizumi@ntt.com

Abstract

Contrastive language-audio pre-training (CLAP) enables zero-shot (ZS) inference of audio and exhibits promising performance in several classification tasks. However, conventional audio representations are still crucial for many tasks where ZS is not applicable (e.g., regression problems). Here, we explore a new representation, a general-purpose audio-language representation, that performs well in both ZS and transfer learning. To do so, we propose a new method, M2D-CLAP, which combines self-supervised learning Masked Modeling Duo (M2D) and CLAP. M2D learns an effective representation to model audio signals, and CLAP aligns the representation with text embedding. As a result, M2D-CLAP learns a versatile representation that allows for both ZS and transfer learning. Experiments show that M2D-CLAP performs well on linear evaluation, fine-tuning, and ZS classification with a GTZAN state-of-the-art of 75.17%, thus achieving a general-purpose audio-language representation.

Index Terms: general-purpose audio-language representation, masked modeling duo, CLIP, CLAP

1. Introduction

The advent of CLIP [1] has had a significant impact on diverse domains and promoted the introduction of various audio-language models (ALMs) in the audio domain [2, 3, 4, 5, 6]. These ALMs have enabled diverse applications, including zero-shot (ZS) classification and audio-to-text/text-to-audio retrieval.

On the other hand, conventional audio models (AMs) and their audio representations also remain essential for tasks that cannot be solved with language. For example, it is challenging to represent the continuous values that are the prediction targets of regression problems in language. CLAP training data are unlikely to contain the language representations of sounds that appear in specific domains, such as industry and medicine. Therefore, the task that ZS classification can solve is limited.

This study explores a general-purpose audio-language representation as a new representation that can serve as both an ALM and a conventional AM. When used as a conventional AM, the representation can serve as audio features for a wide range of tasks, including regression, and when used as an ALM, it serves for various tasks, such as ZS classification.

To achieve this, we propose M2D-CLAP, which combines Masked Modeling Duo [7] (M2D), a self-supervised learning (SSL) method, with learning by CLAP. M2D is an SSL-based AM that uses masked prediction to pre-train a SOTA general-purpose audio representation useful for diverse tasks in transfer learning. Together with CLAP, it enables ZS inference by learning representations that align with textual representations.

Experiments show high transfer learning performance as well as competitive performance in ZS classification, demonstrating that M2D-CLAP achieves a general-purpose audio-language representation. We validate our proposal with many SOTA ALMs/AMs in a unified test environment and compare their performance. Our contributions are i) the introduction of a general-purpose audio-language representation, ii) proposal of M2D-CLAP, and iii) extensive validation of our representation compared to many SOTA models. We also release our code and a new caption dataset for future research[1].

2. Related Work

General-purpose audio representations, proposed in SSL methods such as COLA [8] and BYOL-A [9], have shown effectiveness in various environmental sound, speech, and music tasks. Representations pre-trained by supervised learning methods, such as PANNs [10], AST [11], and HTS-AT [12], have also shown general-purpose effectiveness in various tasks. Masked prediction SSL methods have recently shown remarkable performance: SSAST [13], MAE-AST [14], and MSM-MAE [15] learn through reconstruction tasks, while M2D learns by predicting the representation of masked parts of the input signal. Methods based on masked prediction have also shown high performance: BEATs [16] predict tokenized labels, ATST [17] incorporates data augmentations, and CED [18] distills pre-trained models. While these AM methods are effective in transfer learning, they cannot be applied to ZS classification.

Following CLIP [1], ALM methods capable of ZS audio classification have been actively proposed. AudioCLIP [2] and Wav2CLIP [3] learn audio features that align with the trained CLIP multimodal embedding space. CLAP [4, 5], LAION-CLAP [6], WavCaps [19], and FLAP [20] take an approach similar to CLIP, wherein a variety of audio-caption pair datasets are used to learn text and audio embedding that align. LTU [21], LTU-AS [22], and Pengi [23] take a generative approach using large-language models. These do not have sufficient general-purpose performance, as shown by the experiments in this study.

Approaches similar to the one in this study are SLIP [24] in the image domain, which combines SSL and CLIP, and FLAP, which combines MAE [25] and CLAP. MAE-based SupMAM-CLAP [26] distills CLAP. M2D-S [27] extends M2D with an extra network for speech. Unlike the approaches presented above, we learn general-purpose audio-language representations, ready for both transfer and ZS learning.

3. Proposed Method

We propose M2D-CLAP that learns general-purpose audio-language representations by combining SSL (M2D) and supervised learning (CLAP).

[1] https://github.com/nttcslab/m2d/tree/master/clap

10.21437/Interspeech.2024-29

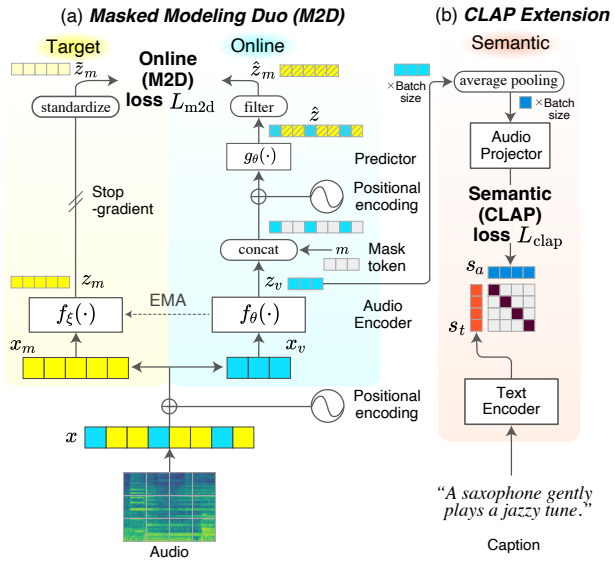

Figure 1: *The M2D-CLAP pre-training flow.*

3.1. Background: Masked Modeling Duo

M2D is a self-supervised learning framework applicable to 2D structured data input such as images and audio spectrograms, and it trains Vision Transformer [28] (ViT) with masked prediction. As shown in Fig. 1(a), it consists of two networks, the online and the target, and learns to predict the target output representations using the online output representations. M2D takes a spectrogram (e.g., 80 frequency bins and 608 time steps) as the input x, which is split into patches (e.g., 16×16) and treated as a series (e.g., $(80/16) \times (608/16) = 190$ patches). M2D then adds positional encoding to patches and randomly selects a number of patches according to a masking ratio as masked patches x_m (e.g., 70% of the input) and the rest as visible patches x_v (e.g., the remaining 30%).

The online network with a set of weights θ encodes x_v using the online encoder f_θ into the representation $z_v = f_\theta(x_v)$. It concatenates the learnable mask tokens m to z_v, adds the position encoding p, and inputs them to the predictor g_θ to predict the representation $\hat{z} = g_\theta(\mathrm{concat}(z_v, m) + p)$. It then outputs the prediction result $\hat{z}_m = \{\, \hat{z}[i] \mid i \in I_M \,\}$ of the masked patch representations, where I_M is the set of masked patch indices.

The target network defined by parameter ξ outputs the representation $z_m = f_\xi(x_m)$ and standardizes it to the final target output $\tilde{z}_m = (z_m - \mathrm{mean}(z_m))/\sqrt{\mathrm{var}(z_m)}$. The loss is calculated using the online prediction $\hat{z}_m$ against the target output $\tilde{z}_m$ as a training signal by the mean square error (MSE) of l_2-normalized $\hat{z}_m$ and $\tilde{z}_m$:

$$L_{\mathrm{m2d}} \triangleq ||l_2(\hat{z}_m) - l_2(\tilde{z}_m)||_2^2 = 2 - 2 \cdot \frac{\langle \hat{z}_m, \tilde{z}_m \rangle}{||\hat{z}_m||_2 \cdot ||\tilde{z}_m||_2}, \quad (1)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

The M2D framework updates θ only to minimize the loss L_{m2d} as depicted by the stop-gradient in Fig. 1(a), and updates $\xi \leftarrow \alpha\xi + (1-\alpha)\theta$ as an exponential moving average of θ with a decay rate α. M2D exploits the momentum encoder to learn effective representations from the target network.

3.2. M2D-CLAP

M2D-CLAP performs a multitask learning of M2D and CLAP by adding the CLAP extension shown in Fig. 1(b) to M2D; M2D-CLAP takes an audio-caption pair as input, feeding audio to M2D and captions to the semantic network in the CLAP extension. It learns from both the online loss from M2D's masked prediction task and the semantic loss from the CLAP extension.

The semantic network maps audio and captions in a common semantic embedding space. A text encoder converts captions to a d_s dimensional vector sentence embedding s_t, which we use as the semantic embedding as it is. The audio projector in the network averages the audio visible patch embeddings z_v encoded by M2D and maps them to a d_s dimensional vector semantic embedding s_a.

The semantic loss follows CLAP using the cosine similarity S_{mn} between s_a and s_t:

$$S_{mn} = \frac{\langle s_a^{(m)}, s_t^{(n)} \rangle}{||s_a^{(m)}||_2 \cdot ||s_t^{(n)}||_2}, \quad (2)$$

where $s_a^{(m)}$ is the semantic embedding of the mth audio batch sample, and $s_t^{(n)}$ is the semantic embedding of the nth caption batch sample. The semantic loss L_{clap} is the average of the NT-Xent losses calculated along the audio and caption axes:

$$L_{\mathrm{clap}} = -\frac{1}{2B} \sum_i^B \left[\log \frac{\exp S_{ii}/\tau}{\sum_j^B \exp S_{ji}/\tau} + \log \frac{\exp S_{ii}/\tau}{\sum_j^B \exp S_{ij}/\tau} \right], \quad (3)$$

where B is the number of batch samples, and τ is the learnable temperature parameter. We follow CLIP [1] to initialize τ with 0.07 and clip it to prevent scaling the logits by more than 100 for training stability.

The entire loss L combines L_{m2d} and L_{clap}:

$$L = \lambda_{\mathrm{m2d}} L_{\mathrm{m2d}} + \lambda_{\mathrm{clap}} L_{\mathrm{clap}}, \quad (4)$$

where the loss weights λ_{m2d} and λ_{clap} control the contribution. After pre-training, we transfer only the audio encoder and projector to downstream tasks; the encoder output is for transfer learning, and the projector output is for ZS learning.

Unlike other methods, sentence embedding is used as a multimodal common semantic embedding space to map the audio embedding. This is beneficial for training an audio embedding to align with the existing semantic embedding space when the space is rich or versatile enough to be compatible with other modalities such as images, for example. In our experiments, we used General Text Embeddings [29] (GTE) with fixed weights as a text encoder and an MLP as an audio projector.

4. Experiments

We validate our method in the scenarios of transfer learning by linear evaluation (Section 4.3), fine-tuning (Section 4.4), and ZS learning (Section 4.5).

4.1. Training Dataset

We used AudioSet [30] audio data to train M2D-CLAP, as in M2D. It consists of 2,005,132 samples (5569 h) of 10-s audio from the balanced and unbalanced train segments.

To form paired audio-caption data with AudioSet, we used the large-scale caption dataset Auto-ACD [31] as the primary data and our newly created caption dataset, AudioCaps Alternative 4 Captions (ACalt4), to provide variations. Auto-ACD consists of over 1.9M captions. We used the AudioSet subset of Auto-ACD, created label-based captions *"The sound of ⟨labels⟩"* for the missing captions in Auto-ACD, and made a complete paired dataset of our copy of AudioSet.

ACalt4 is another variation of the AudioCaps [32] caption dataset (45K samples) for the audio samples in the subset of the

Table 1: *Fine-tuning settings*

Parameter	AS2M	AS20K	ESC-50	SPCV2	VC1
Learning rate	2.0	0.5	0.5	0.5	0.0005
Batch size	64	64	128	128	64
Optimizer	LARS	SGD	SGD	SGD	AdamW
Mixup ratio	0.5	0.3	0.0	0.3	0.0
Random resize crop (RRC)	-	✓	✓	✓	-
SpecAugment[♯] [34]	30/192	30/192	15/48	30/48	30/48
Training epochs (total)	70	200	200	200	50
Training epochs (warm-up)	15	5	5	5	5
Structured Patchout [35] ratio	0.5	0.5	0.5	0.5	0.0

[♯] The frequency/time masking parameters.

Table 2: *ZS caption conversion rules*

Task	Rule
AS & FSD	*"⟨labels⟩ can be heard"*
ESC50 & US8K	*"⟨label⟩ can be heard"*
CREMA-D	*"(angry person talking | someone talking in disgust | someone talking with a sense of fear | someone talking happily and joyfully | someone talking calmly | someone talking sadly) can be heard"*
GTZAN	*"⟨label⟩ music can be heard"*
NSynth	*"the musical instrument sound of ⟨label⟩ can be heard"*

AudioSet. ACalt4 provides four captions for each of the 41,785 samples. To build this dataset, we used the images extracted from the video of the AudioSet sample, as in Auto-ACD. We generated the captions through an automatic pipeline that inputs the image captions generated by BLIP-2 [33], and the AudioSet labels and formats them by leveraging ChatGPT[2].

4.2. Experimental Setup

We used the same M2D configurations as in [7], including the use of ViT Base as the encoder, with an input audio duration of 6 s and a fixed masking ratio of 0.7. For a sentence encoder, we used GTE-base [29] from Hugging Face[3], with the feature dimension d_s of 768, and fixed the weights. The audio projector is a two-layer MLP with a hidden size of 768.

For the audio data, we randomly cropped 6-s audio from a 10-s sample. We preprocessed audio samples to a log-scaled mel spectrogram with a sampling frequency of 16,000 Hz, window size of 25 ms, hop size of 10 ms, and mel-spaced frequency bins $F = 80$ in the range of 50 to 8000 Hz and standardized them with the statistics of AudioSet.

This study differs from [7] in that we use the statistics from our pre-training with AudioSet when standardizing the spectrograms for each downstream task. Specifically, we used an average of -7.1 and a standard deviation of 4.2 throughout all downstream task evaluations. We conducted all evaluations using EVAR[4] as a unified evaluation platform.

Pre-training details We followed M2D for all pre-training settings, including a batch size of 2048 and training epochs of 300. The loss weights for M2D-CLAP were set to 1.0 for L_{m2d} and 0.01 for L_{clap}. In cases where ACalt4 had captions for an audio sample, five captions were available, one of which was randomly picked for each training step. Unlike other ALM methods that initialize an audio encoder with pre-trained weight parameters, we pre-trained M2D from scratch.

Linear evaluation details All evaluation details and downstream tasks are the same as in [7, 9]. Tasks include ESC-50 [36], UrbanSound8K [37] (US8K), Speech Commands V2 [38] (SPCV2), VoxCeleb1 [39] (VC1), VoxForge [40] (VF), CREMA-D [41] (CRM-D), GTZAN [42], NSynth [43], and Pitch Audio Dataset (Surge synthesizer) [44]. All the tasks are classification problems, and all the results are accuracies.

Fine-tuning details All downstream tasks are the same as in [7]. Tasks include ESC-50, SPCV2, and VC1, plus full AudioSet (AS2M) and the subset AudioSet20K (AS20K). We extended the fine-tuning settings from [7]. In addition to Mixup [9, 45], RRC [9], and Structured Patchout [35], we used SpecAugment [34] for data augmentation. The positional encoding was interpolated to adjust it to the duration of the audio sample of the task for AS2M, AS20K, and VC1. The patch embedding layer weights in ViT were fixed to stabilize the fine-tuning [46] for ESC-50. Table 1 summarizes the settings.

ZS evaluation details The ZS tasks include AudioSet (AS), ESC-50 (ESC), US8K, CREMA-D (CRD), GTZAN (GTZ), NSynth (NS), and a multi-label classification FSD50K [47] (FSD). We conducted the ZS classification in the standard procedure. The model's prediction result was obtained as the label with the closest cosine distance between each test sample and the label's caption, and we obtained the accuracy using these prediction results. Table 2 summarizes the conversion rule of the captions from task labels.

4.3. Evaluating Frozen Models (Linear Evaluation)

We evaluated the SOTA audio and audio-language models and the baseline audio models MSM-MAE and M2D. Note that the evaluation was conducted under a unified platform with publicly available pre-trained weights, as in [7] and [9], for fair comparison. The baselines MSM-MAE and M2D were evaluated under the same conditions described in Section 4.2.

The experimental results in Table 3 show that M2D-CLAP performs best on two tasks and has the best average results, demonstrating that it is effective as a general-purpose representation. Compared to the baselines, the performance is significantly improved for ESC-50 and NSynth, indicating the effect of learning from the caption's supervision. However, performance deteriorates by about 3pp for VC1 (1251 speaker identification), and we confirmed in preliminary experiments that performance drops further with larger λ_{clap}, suggesting a trade-off between the CLAP and M2D learnings. Notably, M2D-CLAP performed well on Surge, an 88 MIDI note classification task similar to a regression problem that is tough for ZS inference to solve. Overall, this experiment validates that M2D-CLAP retains high general-purpose performance with its frozen representations.

Results also show that the performance of ALM representations varies from task to task and is not generally effective. While ESC-50, US8K, GTZAN, and Nsynth show high performance, the other five tasks show low performance, especially VC1, which is less than 20% compared to over 70% of the top performance. This may indicate that the coverage of linguistic expressions in the current captions is still limited. Overall, the ALMs' representations underperform the top results by more than 10pp on average, and they are thus considered to be less versatile as frozen representations.

4.4. Evaluating Fine-tuning Performance

Table 4 shows the results of fine-tuning. Unlike in the other experiments, we obtained only the results for the baseline and our models due to the difficulty of reproducing the results of other methods in fine-tuning. M2D-CLAP improved results for AS2M, AS20K, and ESC-50. Meanwhile, it degraded VC1 performance, showing the same trend as in the linear evaluation. However, the performance of VC1 is similar to that of ATST-Clip, indicating that M2D-CLAP retains its general-purpose performance in the results. Notably, M2D-CLAP requires only

[2] https://openai.com/chatgpt

[3] https://huggingface.co/thenlper/gte-base

[4] https://github.com/nttcslab/eval-audio-repr

Table 3: *Linear evaluation results (%) with 95% CI. We evaluated all models under a unified condition except Pengi.*

	Env. sound tasks		Speech tasks				Music tasks			
Model (/masking ratio)	ESC-50	US8K	SPCV2	VC1	VF	CRM-D	GTZAN	NSynth	Surge	Avg.
(Previous studies: Audio models)										
CED [18]	97.3 ±0.5	87.8 ±0.2	89.0 ±0.3	35.2 ±0.2	94.8 ±0.1	66.1 ±1.3	42.3 ±15.4	75.6 ±0.5	38.9 ±0.6	69.7 ±2.1
$\text{BEATs}_{\text{iter3}}$ [16]	86.9 ±1.4	84.8 ±0.1	89.4 ±0.1	41.4 ±0.7	94.1 ±0.3	64.7 ±0.8	72.6 ±4.3	75.9 ±0.2	39.3 ±0.4	72.1 ±0.9
$\text{BEATs}_{\text{iter3+}}$ [16]	95.5 ±0.3	87.6 ±0.3	86.7 ±0.1	37.0 ±0.2	92.5 ±0.1	67.6 ±1.5	84.6 ±0.5	73.1 ±0.4	35.7 ±0.3	73.4 ±0.4
ATST-Clip [17]	94.1 ±0.6	85.8 ↰	95.1 ↰	72.0 ↰	97.6 ±0.0	68.8 ±1.3	78.9 ±3.5	76.2 ↰	32.8 ±0.0	77.9 ±1.1
ATST-Frame [17]	90.9 ±0.6	85.8 ↰	94.9 ↰	**77.4** ↰	**98.8 ±0.3**	72.3 ±0.7	82.9 ±6.0	75.9 ↰	40.6 ±0.2	79.9 ±1.6
HTS-AT [12]	95.7 ±0.7	83.8 ±0.1	82.1 ±0.3	18.1 ±0.4	82.3 ±0.3	56.2 ±0.6	**85.1 ±0.5**	73.3 ±0.8	26.3 ±0.5	67.0 ±0.5
(Previous studies: Audio-Language models)										
LAION-CLAP [6]	97.3 ±0.5	86.9 ±0.5	75.9 ±0.5	13.4 ±0.4	80.3 ±0.2	54.6 ±1.0	84.3 ±2.6	72.2 ±1.1	14.8 ±0.5	64.4 ±0.8
CLAP_{2022} [4]	93.8 ±0.1	84.2 ±0.7	59.0 ±1.1	8.9 ±0.6	75.8 ±1.3	54.4 ±0.8	79.3 ↰	68.2 ±0.6	8.4 ±0.7	59.1 ±0.7
CLAP_{2023} [5]	**97.7 ±0.5**	88.4 ±0.1	86.2 ±0.8	21.1 ±0.3	89.6 ±0.8	62.5 ±1.8	82.3 ±0.5	**80.5 ±0.1**	27.2 ±0.5	70.6 ±0.6
Pengi [23]	89.15 ↰	-	-	-	-	50.57 ↰	80.0 ↰	-	-	-
WavCaps [19]	97.2 ±0.3	63.6 ±0.6	73.3 ±1.7	16.9 ±0.2	80.0 ±1.0	58.6 ±0.7	80.2 ±1.3	74.4 ±0.9	21.1 ±0.2	62.8 ±0.8
(Baseline: Audio models)										
MSM-MAE/0.75 [15] †	89.2 ±0.9	87.4 ±0.2	96.0 ±0.1	73.6 ±0.2	97.8 ±0.2	71.2 ±0.4	79.2 ±0.9	74.6 ±0.9	**43.3 ±0.3**	79.1 ±0.5
M2D/0.6 [7] †	91.6 ±0.5	87.2 ±0.3	**96.2 ±0.1**	75.0 ±0.3	98.2 ±0.1	71.4 ±0.9	83.4 ±3.6	76.1 ±0.1	41.7 ±0.2	80.1 ±0.7
M2D/0.7 [7] †	91.3 ±0.6	87.6 ±0.2	96.0 ±0.1	73.4 ±0.2	98.3 ±0.0	73.0 ±0.7	84.1 ±2.7	75.7 ±0.1	42.1 ±0.2	80.2 ±0.5
(Ours: Audio-Language model)										
M2D-CLAP/0.7	96.3 ±0.3	**88.8 ±0.6**	95.8 ±0.3	70.3 ±0.4	98.3 ±0.1	**73.4 ±0.2**	84.1 ±1.5	78.0 ±0.5	42.4 ±0.6	**80.8 ±0.5**

↰ Results quoted from corresponding papers when they are better than ours or unavailable in our test.
† Results obtained with the experimental setup in Section 4.2.

Table 4: *Fine-tuning results with 95% CI. All results of previous studies are quoted from corresponding papers.*

Model (/masking ratio)	AS2M mAP	AS20K mAP	ESC-50 acc(%)	SPCV2 acc(%)	VC1 acc(%)
(Previous studies: Audio models)					
CED [18] ✕	**50.0**	**44.0**	96.65	-	-
$\text{BEATs}_{\text{iter3}}$ [16]	48.0	38.3	95.6	98.3	-
$\text{BEATs}_{\text{iter3+}}$ [16] ♯	48.6	41.8	**98.1**	98.1	-
SupMAM-CLAP [26] ♯	48.5	38.6	97.6	**98.7**	-
ATST-Clip [17]	45.2	37.9	-	98.0	95.5
ATST-Frame [17]	48.0	39.0	-	98.1	97.3
ATST-C2F [17] ♯	49.7	40.5	-	98.4	**97.5**
HTS-AT [12]	47.1	-	97.0	98.0	-
(Previous studies: Audio-Language models)					
AudioCLIP [2]	-	-	97.15	-	-
Wav2CLIP [3]	-	-	85.95	-	-
CLAP_{2022} [4]	-	-	96.7	96.8	-
(Baseline: Audio models)					
MSM-MAE/0.75 [15] †	47.4 ±0.1	37.9 ±0.0	95.4 ±0.1	98.4 ±0.0	96.6 ±0.1
M2D/0.6 [7] †	47.7 ±0.2	38.4 ±0.1	95.6 ±0.1	98.5 ±0.1	96.5 ±0.1
M2D/0.7 [7] †	47.9 ±0.0	38.6 ±0.1	96.0 ±0.2	98.4 ±0.1	96.3 ±0.2
(Ours: Audio-Language model)					
M2D-CLAP/0.7	48.5 ±0.1	41.8 ±0.2	97.4 ±0.2	98.3 ±0.1	95.5 ±0.2

♯ Results using multiple pre-trainings/objectives or ✕ large models to distill.
† Results obtained with the experimental setup in Section 4.2.

Table 5: *ZS classification results. Underlined results used test task data during training (≠ a ZS scenario).*

Model	AS mAP	FSD mAP	ESC acc(%)	US8K acc(%)	CRD acc(%)	GTZ acc(%)	NS acc(%)
AudioCLIP [2]	-	-	69.40 ↰	68.78 ↰	-	-	-
Wav2CLIP [3]	-	3.02 ↰	41.4 ↰	40.44 ↰	-	-	-
WavCaps [19]	19.60	**52.96**	94.8 ↰	81.42	19.86	45.52	27.66
LAION-CLAP [6]	-	45.85	91.0 ↰	77.0 ↰	23.08	47.24	**35.28**
Proto-LC [48]	-	52 ↰	**96** ↰	73 ↰	-	-	-
CLAP_{2022} [4]	**5.8** ↰	30.24 ↰	82.6 ↰	75.29	22.76	28.97	21.44
CLAP_{2023} [5]	10.2 ↰	48.5 ↰	93.90 ↰	**82.3** ↰	**30.0** ↰	58.4 ↰	**58.08**
Pengi [23]	16.35 ↰	46.76 ↰	91.95 ↰	71.85 ↰	18.46 ↰	35.25 ↰	50.07 ↰
LTU[21]/-AS[22]	18.7 ↰	46.3 ↰	83.1 ↰	-	-	50.3 ↰	-
JMLA [49]	-	-	-	-	-	64.82 ↰	-
M2D-CLAP/0.7	**27.24**	**40.82**	75.45	72.40	17.73	**75.17**	23.39

↰ Results quoted from each paper when they are better than ours or unavailable in our test.

a single pre-training to achieve competitive results with SOTA methods involving multi-iteration/model pre-training.

Among the previous ALMs, CLAP's SPCV2 result of 96.8% underperforms AMs' 98%+. However, the performance gap is much smaller than that of linear evaluation, indicating a modest effectiveness of ALMs' representations in fine-tuning.

4.5. Evaluating ZS Classification Performance

Table 5 shows the ZS classification results. M2D-CLAP performs poorly on ESC-50 but well on AudioSet and GTZAN. Particularly, it updates the SOTA result on GTZAN. Although not in a valid ZS scenario, the best performance on AudioSet is likely because it is the only model trained on AudioSet alone among the ones with AS results. That also explains the poor performance on ESC-50; CLAP [4] reports that their ESC-50 performance has dropped from 82.6% to 67.15% by adding the 1.7M AudioSet samples, aligning with our result of 75.45 %. Overall, M2D-CLAP showed competitive ZS performance.

5. Conclusion

This study explored a general-purpose audio-language representation ready for both zero-shot inference and conventional transfer learning. To this end, we proposed M2D-CLAP, which combines CLAP learning with M2D, an SSL method for learning effective general-purpose representations. In our experiments, M2D-CLAP showed high performance in linear evaluation, fine-tuning, and zero-shot classification scenarios, and we confirmed that it could learn the desired general-purpose audio-language representation. In particular, M2D-CLAP further improved the performance of the general-purpose representation compared to M2D and updated GTZAN's SOTA performance in the zero-shot classification. The general-purpose audio-language representation is effective both as an audio-language model and as a conventional audio representation and is expected to be beneficial for many future application tasks. Our code and dataset are available online for future studies[5].

6. References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *ICML*, 2021, pp. 8748–8763.

[2] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending Clip to Image, Text and Audio," in *ICASSP*, 2022, pp. 976–980.

[3] H.-H. Wu, P. Seetharaman, K. Kumar, and J. P. Bello, "Wav2CLIP: Learning Robust Audio Representations from Clip," in *ICASSP*, 2022, pp. 4563–4567.

[4] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "CLAP: Learning Audio Concepts From Natural Language Supervision," in *ICASSP*. IEEE, 2023.

[5] B. Elizalde, S. Deshmukh, and H. Wang, "Natural Language Supervision for General-Purpose Audio Representations," *arXiv preprint arXiv:2309.05767*, 2023.

[6] 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 *ICASSP*, 2023.

[5] https://github.com/nttcslab/m2d/tree/master/clap

[7] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked Modeling Duo: Learning Representations by Encouraging Both Networks to Model the Input," in *ICASSP*, 2023.

[8] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *ICASSP*, 2021, pp. 3875–3879.

[9] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "BYOL for Audio: Exploring Pre-trained General-purpose Audio Representations," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 31, p. 137–151, 2023.

[10] 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, Language Process.*, vol. 28, pp. 2880–2894, 2020.

[11] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Interspeech*, 2021, pp. 571–575.

[12] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *ICASSP*, 2022, pp. 646–650.

[13] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, "SSAST: Self-Supervised Audio Spectrogram Transformer," in *AAAI*, vol. 36, no. 10, 2022, pp. 10 699–10 709.

[14] A. Baade, P. Peng, and D. Harwath, "MAE-AST: Masked Autoencoding Audio Spectrogram Transformer," in *Interspeech*, 2022, pp. 2438–2442.

[15] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked Spectrogram Modeling using Masked Autoencoders for Learning General-purpose Audio Representation," in *HEAR (NeurIPS 2021 Competition)*, vol. 166, 2022, pp. 1–24.

[16] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "BEATs: Audio Pre-Training with Acoustic Tokenizers," in *ICML*, 2023.

[17] X. Li, N. Shao, and X. Li, "Self-Supervised Audio Teacher-Student Transformer for Both Clip-Level and Frame-Level Tasks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 32, pp. 1336–1351, 2024.

[18] H. Dinkel, Y. Wang, Z. Yan, J. Zhang, and Y. Wang, "CED: Consistent ensemble distillation for audio tagging," in *ICASSP*, 2024.

[19] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "WavCaps: A ChatGPT-Assisted Weakly-Labelled Audio Captioning Dataset for Audio-Language Multimodal Research," *arXiv preprint arXiv:2303.17395*, 2023.

[20] C.-F. Yeh, P.-Y. Huang, V. Sharma, S.-W. Li, and G. Gosh, "FLAP: Fast Language-Audio Pre-Training," in *ASRU*, 2023.

[21] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. Glass, "Listen, Think, and Understand," in *ICLR*, 2024.

[22] Y. Gong, A. H. Liu, H. Luo, L. Karlinsky, and J. Glass, "Joint Audio and Speech Understanding," in *ASRU*, 2023.

[23] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An Audio Language Model for Audio Tasks," in *NeurIPS*, 2023.

[24] N. Mu, A. Kirillov, D. Wagner, and S. Xie, "SLIP: Self-supervision Meets Language-Image Pre-training," in *ECCV*, 2022, pp. 529–544.

[25] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *CVPR*, 2022.

[26] Y. Xin, X. Peng, and Y. Lu, "Masked Audio Modeling with CLAP and Multi-Objective Learning," in *Interspeech*, 2023, pp. 2763–2767.

[27] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked Modeling Duo for Speech: Specializing General-Purpose Audio Representation to Speech using Denoising Distillation," in *Interspeech*, 2023, pp. 1294–1298.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[29] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, "Towards general text embeddings with multi-stage contrastive learning," *arXiv preprint arXiv:2308.03281*, 2023.

[30] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017, pp. 776–780.

[31] L. Sun, X. Xu, M. Wu, and W. Xie, "A large-scale dataset for audio-language representation learning," *arXiv preprint arXiv:2309.11500*, 2023.

[32] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating Captions for Audios in The Wild," in *NAACL-HLT*, 2019.

[33] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in *ICML*, 2023.

[34] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Interspeech*, 2019, pp. 2613–2617.

[35] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *Interspeech*, pp. 2753–2757, 2022.

[36] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *ACM-MM*, 2015, pp. 1015–1018.

[37] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM-MM*, 2014, pp. 1041–1044.

[38] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *arXiv preprint arXiv::1804.03209*, Apr. 2018.

[39] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Interspeech*, 2017, pp. 2616–2620.

[40] K. MacLean, *"Voxforge"*, 2018, available at http://www.voxforge.org/home.

[41] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-D: Crowd-sourced emotional multimodal actors dataset," *IEEE Trans. Affective Comput.*, vol. 5, no. 4, 2014.

[42] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Speech Audio Process.*, vol. 10, no. 5, 2002.

[43] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *ICML*, 2017.

[44] J. Turian, J. Shier, G. Tzanetakis, K. McNally, and M. Henry, "One billion audio sounds from GPU-enabled modular synthesis," in *DAFx2020*, 2021.

[45] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[46] A. Kumar, R. Shen, S. Bubeck, and S. Gunasekar, "How to Fine-Tune Vision Models with SGD," *arXiv preprint arXiv:2211.09359*, 2022.

[47] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An Open Dataset of Human-Labeled Sound Events," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 829–852, 2022.

[48] S. S. Kushwaha and M. Fuentes, "A multimodal prototypical approach for unsupervised sound classification," in *Interspeech*, 2023, pp. 266–270.

[49] X. Du, Z. Yu, J. Lin, B. Zhu, and Q. Kong, "Joint Music and Language Attention Models for Zero-shot Music Tagging," *arXiv preprint arXiv:2310.10159*, 2023.