Abstract

Self-supervised learning general-purpose audio representations have demonstrated high performance in a variety of tasks. Although they can be optimized for application by fine-tuning, even higher performance can be expected if they can be specialized to pre-train for an application. This paper explores the challenges and solutions in specializing general-purpose audio representations for a specific application using speech, a highly demanding field, as an example. We enhance Masked Modeling Duo (M2D), a general-purpose model, to close the performance gap with state-of-the-art (SOTA) speech models. To do so, we propose a new task, denoising distillation, to learn from fine-grained clustered features, and M2D for Speech (M2D-S), which jointly learns the denoising distillation task and M2D masked prediction task. Experimental results show that M2D-S performs comparably to or outperforms SOTA speech models on the SUPERB benchmark, demonstrating that M2D can specialize in a demanding field.

Index Terms: speech representation learning, general-purpose audio representation, denoising, distillation, specialization

1. Introduction

The self-supervised learning general-purpose audio representations (generic models), pre-trained on a large-scale audio dataset, have shown promising performance on a variety of environmental, musical, and speech tasks. While the generic models have demonstrated versatile performance [1, 2, 3, 4, 5], they have not fully demonstrated their usefulness in focused applications such as automatic speech recognition, where SOTA performance is required.

On the other hand, highly advanced speech models have been investigated to meet intense demand. In addition to effective self-supervised learning using contrastive loss and masked prediction [6, 7, 8, 9], SOTA models make use of the learning of discrete representations by quantization [7], pseudo-label generation by clustering [8, 9], and denoising for learning robust representations [9].

We believe generic models can provide even higher performance when pre-trained specifically for a particular application rather than just fine-tuning for the application. This paper explores the challenges and possibilities of specialization using a generic model, masked modeling duo (M2D) [4], in speech as an example. The question we aim to answer is: Can a general-purpose audio representation be useful in a competitive field?

To specialize a generic model in speech, we found that the challenges lie in incorporating speech-specific techniques. While a generic model can easily switch the pre-training dataset to a speech corpus, we found it is not yet powerful enough to compete with SOTA speech models. Therefore, resorting to field-specific extensions (e.g., learning from clustered features) is inevitable.

With a necessary extension, we propose M2D for Speech (M2D-S), which extends M2D with a new task, denoising distillation, to learn from fine-grained clustered features. M2D-S also adapts the pre-training dataset and patch size without changing M2D. Experiments using the SUPERB benchmark [10] show that M2D-S performs comparably to or better than SOTA speech models, demonstrating that a generic model can serve as a pre-training framework for a challenging field.

Our contributions are i) proposal of a denoising distillation task for speech representation learning, ii) proposal of M2D-S to specialize a generic model to speech, and iii) demonstration of the extensibility of a generic model by achieving SOTA performance in speech. Our code is available online1.

2. Related Work

Self-supervised learning methods that train transformers with masked prediction have shown promising performance in various domains. Speech representation models, such as Mockingjay [11] and TERA [12], take spectrograms as input. TERA employs a masking strategy for splitting both frequency bins and time steps, similar to generic models. SOTA models, such as wav2vec2.0 [7], BigSSL [13], data2vec [14], HuBERT [8], and WavLM [9], typically learn an acoustic feature extraction from the input speech waveform in addition to the representation learning. Notably, models such as wav2vec2.0, HuBERT, and WavLM effectively learn by using discretized pseudo-labels with vector quantization or clustering of pre-trained model features. In addition, WavLM has improved performance on non-
ASR tasks through additional masked speech denoising.

Generic models, such as AST [15], SSAST [1], ATST [16], MAE-AST [2], MSM-MAE [3], AudioMAE [17], M2D [4], and BEATs [5], have shown SOTA performance. They typically take spectrograms as acoustic feature input, split input in both the time and frequency axes, and train a Vision Transformer (ViT) [18] while avoiding application-specific techniques such as masking inputs for consecutive time steps, as in speech.

In previous works similar to this paper, Melms et al. [19] and BYOL-S [20] have specialized a generic model BYOL-A [21] in medical and speech applications. SSAST has adapted patch size and pre-training dataset and compared it with speech models on SUPERB. In the NLP domain, LIBERT [22] specializes BERT [23] using an additional task for pre-training a lexically-informed BERT, a similar multi-task learning setting to ours. Previous works that created pseudo-labels or distill models include BEATs, HuBERT, WavLM, DistilHuBERT [24], MT4SSL [25], and RobustDistiller [26]. In particular, concurrent works MT4SSL and RobustDistiller learn multi-tasks, similar to ours, and RobustDistiller auxiliary learns to denoise speech in addition to distillation.

Unlike these previous studies, we attempt to achieve the performance of SOTA speech models by specializing M2D.

3. Method

M2D-S specializes in speech by adding a speech extension network and its task to M2D without changing M2D. Table 1 lists the major differences between M2D and SOTA speech models. The key challenge of speech specialization is to implement in M2D-S the features that critically impact speech task performance.

### Table 1: Key differences between models.

<table>
<thead>
<tr>
<th>Design choices</th>
<th>Speech model</th>
<th>Generic model</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Pre-training dataset</td>
<td>Speech (e.g., LibriSpeech)</td>
<td>General audio (e.g., HUGBERT, WavLM)</td>
</tr>
<tr>
<td>(b) Input data format</td>
<td>Raw waveform</td>
<td>Spectrogram</td>
</tr>
<tr>
<td>(c) Input data split</td>
<td>Split in time steps</td>
<td>Split in both freq/time</td>
</tr>
<tr>
<td>(d) Feature extractor</td>
<td>CNN</td>
<td>(not used)</td>
</tr>
<tr>
<td>(e) Discretized</td>
<td>Clustering pre-trained (not used)</td>
<td>training signals (model features)</td>
</tr>
<tr>
<td>(f) Extra pre-training task</td>
<td>Denoising (WavLM) (not used)</td>
<td>Masking strategy</td>
</tr>
<tr>
<td>(g) Masking strategy</td>
<td>Consequent time steps</td>
<td>Random</td>
</tr>
</tbody>
</table>

Based on the preliminary experimental results, we focus on (a) using a speech corpus as the dataset, (c) splitting the input only along the time steps, (e) using clustered pre-trained model features as a training signal, and (f) using a denoising task.

In addition, we propose a new task, denoising distillation, which maximizes the use of fine-grained clusters in the distribution of the pre-trained speech model features and performs (e) and (f) simultaneously.

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 trains ViT with masked prediction. As shown in Fig. 2(a), it consists of two networks, the online and the target, and learns to predict the target output representations using the online output representations. Unlike speech models (e.g., HuBERT), M2D takes a spectrogram (e.g., 80 frequency bins and 208 time steps) as input, which is split into patches (e.g., 16 × 16) and treated as a series (e.g., 80/16 × 208/16 = 65 patches).

M2D splits the input data $x$ into patches, adds positional encoding, and randomly selects a number of patches according to a masking ratio as masked patches $x_m$ (e.g., 60% of the input) and the rest as visible patches $x_v$ (e.g., the remaining 40%). The online network with a set of weights $\theta$ encodes $x_m$ using the online encoder $f_o$ to the representation $z_m = f_o(x_m)$. It concatenates the learnable masked tokens $m$ to $z_m$, adds the position encoding $p$, and inputs to the predictor $g_p$ to predict the representation $\hat{z} = g_p(\text{concat}(z_m, m) + p)$. It then outputs the prediction result $\hat{z}_m = \{ \hat{z}[i] | i \in FM \}$ of the masked patch representations, where $FM$ is the set of masked patch indices.

The target network defined by parameter $\xi$ outputs the representation $z_M = f_t(x_m)$ and standardizes it to the final target output $z_m = (z_m - \text{mean}(z_m)) / \sqrt{\text{var}(z_m)}$.

The loss is calculated using the online prediction $\hat{z}_m$ against the target output $z_m$ as a training signal by the mean square error (MSE) of $l_2$-normalized $\hat{z}_m$ and $z_m$:

$$L_{\text{m2d}} \triangleq \| I_2(\hat{z}_m) - I_2(z_m) \|^2_2 = 2 - 2 \cdot \frac{(\| z_m \|_2^2 \cdot \| \hat{z}_m \|_2^2)}{\| z_m \|_2^2 \cdot \| \hat{z}_m \|_2^2},$$

where $(\cdot, \cdot)$ denotes the inner product.

The M2D framework updates parameters $\theta$ only by minimizing the loss $L_{\text{m2d}}$ as depicted by the stop-gradient in Fig. 2 (a) and updates $\xi \leftarrow \tau \xi + (1 - \tau) \theta$ as an exponential moving average of $\theta$ with a decay rate $\tau$.

M2D exploits the momentum encoder to learn effective representations from the target network. After the training, only the $f_o$ is used as a pre-trained model in downstream tasks.

3.2. Denoising Distillation

We think that a model pre-trained with pseudo labels generated by clustering (e.g., k-means) outputs fine-grained clustered features; therefore, using the pre-trained model features as a training signal should be desirable. HuBERT and WavLM create training signals for the next training iteration by clustering the features produced by the previous iteration model. However, since the previous model has already learned to target clustered labels, we think the model should produce clustered features.

That being said, we propose to combine denoising and distillation as a more flexible approach to learning from clustered targets. The distillation method typically forms a teacher-student network and uses the teacher’s output as a training signal to train the student. Inspired by WavLM, we add noise to the student’s input. In this case, the teacher’s output acts as a
microcluster center corresponding to each data sample, and the student learns to predict cluster centers regardless of the noise. As a result, the denoising distillation is expected to learn from fine-grained clustered features.

3.3. Masked Modeling Duo for Speech

Figure 2 shows M2D-S, which adds an offline network as a speech extension to M2D. We use two types of data (speech and background noise): speech only for the speech extension and noisy speech, a mixture of speech and noise, for M2D.

We distill the offline network as a teacher to M2D as a student. In the speech extension, while the offline encoder produces features \( h \), we concatenate \( z_\alpha \) and \( \tilde{z}_\alpha \) from the output of M2D and project them using a linear layer to predict offline encoder features as \( \tilde{h} \). In case M2D splits the input spectrogram in the frequency axis, the \( h \) is calculated to have one feature prediction per frame by concatenating the feature predictions for each frequency belonging to a time frame.

We calculate the offline loss \( L_{\text{off}} \) by the MSE of \( l_2 \)-normalized \( h \) and \( \tilde{h} \), the same as Eq. (1) in M2D:

\[
L_{\text{off}} \triangleq \|l_2(h) - l_2(\tilde{h})\|^2_2 = 2 - \frac{\langle h, \tilde{h} \rangle}{\|h\|_2 \cdot \|\tilde{h}\|_2}
\]

The overall M2D-S loss \( L_{\text{m2dS}} \) is then calculated by combining \( L_{\text{off}} \) and \( L_{\text{off}} \):

\[
L_{\text{m2dS}} = \lambda_{\text{off}} L_{\text{off}} + \lambda_{\text{off}} L_{\text{off}},
\]

where the loss weights \( \lambda_{\text{off}} \) control the contribution. The noisy speech is a mixture of background noise and speech in the \( \alpha \) to \((1 - \alpha)\) ratio, where \( \alpha \) is a dataset noise ratio.

4. Experiments

First, we validate the effect of each modification made in M2D-S in ablation studies of the pre-training dataset (Section 4.2), the input size (Section 4.3), and the pre-training tasks (Section 4.4). Finally, we compare M2D-S combining all the best practices with SOTA models (Section 4.5).

4.1. Experimental Setup

We used the same M2D configurations in M2D-S, including the use of ViT Base as the encoder, with a fixed masking ratio of 0.6. For each experiment, the parameters of interest were varied with the following defaults:\footnote{We provide complete details at: https://github.com/nttcslab/m2d/tree/master/speech
https://huggingface.co/facebook/hubert-base-ls960} dataset noise ratio \( \alpha \) of 0.2, input duration \( T \) of 2.08 s, and patch size of \( 80 \times 4 \). M2D takes log-mel spectrogram features as input, which we preprocessed with a sampling frequency of 16,000 Hz, window size of 25 ms, hop size of 10 ms, and frequency bins \( F \) of 80 in the range of 50 to 8,000 Hz and normalized using dataset statistics.

**Pre-training details** We set the number of epochs to 1,000 and the warm-up epochs to 60, where a single epoch consumes all LibriSpeech samples. All other settings were the same as in the M2D, including the batch size and optimizer settings. We used the 9th transformer layer output of the 2nd-iteration HuBERT Base model\footnote{https://github.com/facebook/hubert-base-ls960} as an offline encoder as in WavLM.

For the speech corpus, we used LibriSpeech with 281,241 samples (960 h) from all training splits (LS-960). For the background noise dataset, we used AudioSet [28] (AS) with 2,005,132 samples (5,569 h) of audio from the balanced and unbalanced train segments. We randomly sampled then randomly cropped a segment for the input duration and mixed it with the speech. When mixing, the log-mel spectrograms were once reverted to a linear scale, mixed according to the ratio, and then converted back to log scale.

**Evaluation details** We evaluated all pre-trained models in SUPERB [10] for the speech task performance and a linear evaluation for the non-speech task. In the SUPERB evaluation, the model weights were frozen, and the weighted sum of the features from all transformer layers was used in the evaluation. The evaluation tasks include phoneme recognition (PR), keyword spotting (KS), intent classification (IC), speaker identification (SID), and emotion recognition (ER).

We used linear evaluation as a supplemental measure to assess non-speech task performance when specializing in speech. The linear evaluation and task details are the same as in BYOL-A [21]. We report the average accuracies of two environmental sound tasks (ENV) and three music tasks (MUS). In this evaluation, we trained a linear layer using only the final layer feature of the frozen model instead of using all layer features as in SUPERB.

### 4.2. Pre-training Dataset Ablations

We assessed the impact of the dataset used for pre-training by comparing the performance when using the target domain dataset, i.e., LibriSpeech, and the general audio dataset, i.e., AudioSet. To do so, we used M2D without speech extension and varied the dataset noise ratio \( \alpha \). When only AudioSet was used, we defined one epoch as the number of LibriSpeech samples, and we randomly sampled from AudioSet.

The results in Table 2 show that the speech tasks perform best when only using LibriSpeech, and performance deteriorates as the ratio of AudioSet increases. Conversely, the non-speech tasks (ENV, MUS) perform best when only using AudioSet, and performance deteriorates as the ratio of LibriSpeech increases. Exceptionally, SID performed best when the ratio \( \alpha = 0.2 \) and there was background noise in the speech. However, the use of a speech domain dataset is generally considered to contribute to speech task performance.

<table>
<thead>
<tr>
<th>Dataset noise ratio ( \alpha )</th>
<th>PR (LS-960 only)</th>
<th>KS (LS-960 only)</th>
<th>IC (LS-960 only)</th>
<th>SID (LS-960 only)</th>
<th>EN (LS-960 only)</th>
<th>MUS (LS-960 only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0 (LS-960 only)</td>
<td>10.98</td>
<td>96.20</td>
<td>95.02</td>
<td>74.77</td>
<td>63.02</td>
<td>66.54</td>
</tr>
<tr>
<td>0.1</td>
<td>11.97</td>
<td>91.94</td>
<td>94.38</td>
<td>77.89</td>
<td>61.44</td>
<td>72.77</td>
</tr>
<tr>
<td>0.2</td>
<td>11.87</td>
<td>93.30</td>
<td>93.28</td>
<td>78.46</td>
<td>61.55</td>
<td>73.57</td>
</tr>
<tr>
<td>0.3</td>
<td>12.19</td>
<td>93.38</td>
<td>94.46</td>
<td>78.25</td>
<td>61.55</td>
<td>74.67</td>
</tr>
<tr>
<td>0.4</td>
<td>12.39</td>
<td>95.15</td>
<td>94.75</td>
<td>77.15</td>
<td>61.52</td>
<td>75.16</td>
</tr>
<tr>
<td>0.5</td>
<td>12.53</td>
<td>93.70</td>
<td>92.14</td>
<td>76.58</td>
<td>61.07</td>
<td>76.38</td>
</tr>
<tr>
<td>1.0 (AudioSet only)</td>
<td>27.04</td>
<td>81.91</td>
<td>82.78</td>
<td>68.54</td>
<td>60.85</td>
<td>83.31</td>
</tr>
</tbody>
</table>
The patch sizes of 40 × 4 and 40 × 2, which assign two patches along the frequency axis, showed the best results for PR and SID, but inferior results on some other tasks. Overall, these results indicate that no single patch split along the frequency axis provides a balanced better performance on the speech tasks. Notably, our results align with SSAST [1].

The input duration results in Table 4 show a trend that the longer the input, the better the results. The trend of better performance when learning from features in longer series is consistent as it is in M2D.

Table 3: M2D patch size ablations. (dataset noise ratio \( \alpha = 0.2 \) and input duration \( T = 2.08s \))

<table>
<thead>
<tr>
<th>Patch size (Freq. × Time)</th>
<th>PR</th>
<th>KS</th>
<th>IC</th>
<th>SID</th>
<th>ER</th>
<th>ENV</th>
<th>MUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 × 16 (M2D [4])</td>
<td>77.92</td>
<td>86.24</td>
<td>83.23</td>
<td>79.65</td>
<td>58.88</td>
<td>80.27</td>
<td>59.60</td>
</tr>
<tr>
<td>40 × 8</td>
<td>29.28</td>
<td>90.25</td>
<td>89.67</td>
<td>77.51</td>
<td>59.39</td>
<td>78.56</td>
<td>56.62</td>
</tr>
<tr>
<td>40 × 4</td>
<td>11.49</td>
<td>91.77</td>
<td>90.93</td>
<td>81.64</td>
<td>60.30</td>
<td>80.00</td>
<td>57.48</td>
</tr>
<tr>
<td>40 × 2</td>
<td>15.14</td>
<td>81.47</td>
<td>82.73</td>
<td>85.44</td>
<td>61.49</td>
<td>80.98</td>
<td>58.67</td>
</tr>
<tr>
<td>80 × 8</td>
<td>30.28</td>
<td>89.69</td>
<td>90.51</td>
<td>75.66</td>
<td>58.97</td>
<td>72.62</td>
<td>54.28</td>
</tr>
<tr>
<td>80 × 4</td>
<td>11.87</td>
<td>93.30</td>
<td>91.28</td>
<td>78.46</td>
<td>61.75</td>
<td>73.57</td>
<td>53.76</td>
</tr>
<tr>
<td>80 × 2 (on speech models)</td>
<td>11.74</td>
<td>93.28</td>
<td>92.51</td>
<td>78.86</td>
<td>60.67</td>
<td>74.98</td>
<td>53.28</td>
</tr>
</tbody>
</table>

Table 4: M2D input duration ablations. (dataset noise ratio \( \alpha = 0.2 \) and patch size 80 × 4)

<table>
<thead>
<tr>
<th>Input duration ( T )</th>
<th>PR</th>
<th>KS</th>
<th>IC</th>
<th>SID</th>
<th>ER</th>
<th>ENV</th>
<th>MUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T=2.08s</td>
<td>11.87</td>
<td>93.30</td>
<td>92.28</td>
<td>78.46</td>
<td>61.75</td>
<td>73.57</td>
<td>53.76</td>
</tr>
<tr>
<td>T=3.04s</td>
<td>9.81</td>
<td>94.99</td>
<td>95.15</td>
<td>79.18</td>
<td>63.39</td>
<td>74.36</td>
<td>52.14</td>
</tr>
<tr>
<td>T=4.00s</td>
<td>8.50</td>
<td>95.04</td>
<td>94.83</td>
<td>81.24</td>
<td>63.81</td>
<td>74.11</td>
<td>52.23</td>
</tr>
<tr>
<td>T=5.12s</td>
<td>8.10</td>
<td>95.24</td>
<td>94.70</td>
<td>78.73</td>
<td>65.47</td>
<td>74.69</td>
<td>51.66</td>
</tr>
<tr>
<td>T=6.08s</td>
<td>7.74</td>
<td>96.46</td>
<td>95.00</td>
<td>80.48</td>
<td>64.07</td>
<td>72.42</td>
<td>50.64</td>
</tr>
</tbody>
</table>

5. Discussion for Future Specialization

This study explored the requirements for specializing M2D in speech, and the lessons learned may be useful for specialization in other acoustic fields.

- Optimizing the patch size may be effective for other applications. While splitting only the time axis was effective for speech, splitting along both the frequency and time axes has been reported as effective for other audio tasks [1, 3].
- In addition to using a dataset from the target domain, combining a denoising task with background noise may be effective in learning application-focused representations.
- Learning the denoising distillation task requires an appropriate teacher model. Although we used a model that learned pseudo-labels by clustering, pre-trained classifier models available in the application field may be effective since they learn from class labels, considered human-curated clusters.

6. Conclusion

This paper explored the challenges and solutions in specializing a general-purpose audio representation (generic model) to a specific domain using speech as an example. We found the challenge is to incorporate speech-specific techniques, such as learning from clustered features.

To achieve speech SOTA performance, we proposed a new task, denoising distillation, to learn from fine-grained clustered features, and M2D for Speech (M2D-S), which jointly learns the denoising distillation task and M2D masked prediction task. In the experiments on the SUPERB benchmark, M2D-S significantly improved the performance of M2D. It performed comparable to SOTA speech models (e.g., HuBERT and WaxLM) and outperformed them in keyword spotting and emotion recognition tasks, demonstrating that a generic model can specialize and be useful in a challenging field.

Our study indicated that a generic model is potentially useful for specializing in other audio applications. Our code is available online for future studies.
7. References


