# IMPROVED DOMAIN GENERALIZATION VIA DISENTANGLED MULTI-TASK LEARNING IN UNSUPERVISED ANOMALOUS SOUND DETECTION

*Satvik Venkatesh*[1,2], *Gordon Wichern*[1], *Aswin Subramanian*[1], *Jonathan Le Roux*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
[2]Interdisciplinary Centre for Computer Music Research, University of Plymouth, UK
satvik.venkatesh@plymouth.ac.uk,{wichern,subramanian,leroux}@merl.com

## ABSTRACT

We investigate a novel multi-task learning framework that disentangles domain-shared features and domain-specific features for domain generalization in anomalous sound detection. Disentanglement leads to better latent features and also increases flexibility in post-processing due to the availability of multiple embedding spaces. The framework was at the core of our submissions to the DCASE2022 Challenge Task 2. We ranked 5th out of 32 teams in the competition, obtaining an overall harmonic mean of 67.57% on the blind evaluation set, surpassing the baseline by 13.5% and trailing the top rank by 3.4%. We also explored machine-specific loss functions and domain generalization methods, which showed improvements on the development set, but were less effective on the evaluation set.

***Index Terms***— Anomaly detection, disentanglement, multi-task learning, domain generalization, representation learning

## 1. INTRODUCTION

Machine condition monitoring using acoustic sensors is an important topic for industry with applications such as factory automation and predictive maintenance. Automatic detection of anomalous sounds is a particularly important application, however, all possible types of anomalous sounds may not be known in advance, and purposefully damaging machinery to collect anomalous sound recordings is undesirable. Thus, there has been much recent research interest in the field of unsupervised anomalous sound detection, where only data collected under normal operating conditions is available for training machine learning models.

Much of the recent progress in unsupervised anomalous sound detection has been driven by DCASE challenges on the topic [1–3]. Typical approaches include those based on autoencoder-like architectures [4–10], where a model trained only on normal data to reconstruct its input should exhibit large reconstruction error when presented with an anomalous example at inference time. Another class of approaches, which we refer to as surrogate task models, use an alternative supervised training task to learn a model of normality, and then measure deviations from normal to predict anomalies. Example surrogate tasks include outlier exposure [6, 11], predicting metadata (e.g., machine instance) or attributes (e.g., operating load) [12–14], and learning to predict what augmentations (e.g., time-stretching or pitch-shifting) were applied to an audio clip [15].

As in many areas where deep learning-based models have become the predominant approach, unsupervised anomalous sound detection suffers from issues related to robustness. To better tackle such issues, the anomalous sound detection tasks of both the 2021 and 2022 DCASE challenges focused on performance under domain shift, where acoustic conditions differ based on environmental background noise or other machine operating conditions. The goal is to develop methods that should perform equally well in a source domain, where most of the (normal) training data comes from, and in a target domain, where only a few normal examples are available. The 2021 challenge task [2] assumed the domain (source or target) of the audio sample was known at inference time (a configuration referred to as domain adaptation), while the 2022 task [3] assumes the domain is unavailable at inference time (a configuration referred to as domain generalization).

While many well-known techniques exist for domain generalization (see [16] for an overview), we focus our efforts on disentangled representation learning [17], where subsets of learned feature dimensions correspond to specific factors in the dataset. Disentanglement has been successfully applied for music information retrieval in the audio domain [18] and in approaches to domain adaptation for image classification [19]. Specifically, we consider learning feature representations for each normal sound example in the training set, where subsets of features are learned using different surrogate tasks. In the case of the DCASE 2022 Task 2 dataset, we learn a subset of *domain-shared* features, whose surrogate task is to predict the section index regardless of domain (each section is dedicated to a specific type of domain shift, with other conditions being shared across domains), and subsets of *domain-specific* features each associated with a surrogate task consisting of predicting a particular machine attribute (e.g., specific states or environmental conditions of the machine), which are typically different across domains and sections. We demonstrate experimentally that our disentangled model performs better than a multi-task learning model where features are not disentangled, and further show that by weighting individual anomaly scores computed over different disentangled dimensions, we obtain an ensemble-like system using a single model. Furthermore, by examining the anomaly score in specific disentangled dimensions, we can better understand which attribute may have caused the anomaly, improving the explainability of deep learning models.

As discussed in our challenge report [20], we also explored machine-specific variations to the loss function. While they led to significant improvements on the dev set, they ultimately performed worse on the eval set. This paper thus focuses on our best performing disentangled models.

This work was performed while S. Venkatesh was an intern at MERL.

## 2. DISENTANGLED ANOMALY DETECTOR

In this paper, we investigate an approach that disentangles a learned latent representation into domain-shared and domain-specific features for domain generalization in anomalous sound detection, as illustrated in Fig. 1. In particular, we refer to sections as domain-shared features and to attributes as domain-specific features. For example, in Fan's *section 00*, machine noises occurring in the source domain are of type W and X, while those occurring in the target domain are of type Y and Z. Therefore, *section 00* is common to both domains but the machine noises are different across domains.

### 2.1. Surrogate Task Training

During training, we have a dataset of $N$ normal training examples for a given machine type, $\mathcal{D} = \{(X^{(n)}, y^{(n)})\}_{n=1}^{N}$, where $X \in \mathbb{R}^{F \times T}$ is a magnitude spectrogram with $F$ frequencies and $T$ time frames, and $y = [y_s, y_{a_1}, \ldots, y_{a_M}] \in \mathbb{N}^{M+1}$ is a vector of categorical surrogate task labels, where $y_s$ represents machine section and $y_{a_m}$ represents the categorical label of the $m$-th attribute among the $M$ different attributes available for the given machine type. We obtain a domain-shared (section) embedding $z_S$ and a domain-specific (attribute) embedding $z_A$ as:

$$z_S = \Phi^{\text{Sec}}[\text{CNN}(X)] \in \mathbb{R}^{D_S}, \quad z_A = \Phi^{\text{Att}}[\text{CNN}(X)] \in \mathbb{R}^{D_A} \tag{1}$$

where $\text{CNN}(\cdot)$ is a shared convolutional neural network, while $\Phi^{\text{Sec}}$ and $\Phi^{\text{Att}}$ represent section and attribute specific linear embedding layers, respectively (implemented as $1 \times 1$ convolutions). All parameters are trained by minimizing $\mathcal{L} = \mathcal{L}^{\text{Sec}} + \mathcal{L}^{\text{Att}}$, where

$$\mathcal{L}^{\text{Sec}} = \log \frac{\exp(w_{0,y_s} \cdot z_S + b_{0,y_s})}{\sum_{c=1}^{C} \exp(w_{0,c} \cdot z_S + b_{0,c})}, \tag{2}$$

$$\mathcal{L}^{\text{Att}} = \sum_{m=1}^{M} \log \frac{\exp(w_{m,y_m} \cdot z_A + b_{m,y_m})}{\sum_{c_m=1}^{C_m} \exp(w_{m,c_m} \cdot z_A + b_{m,c_m})} \tag{3}$$

are the cross-entropy losses for section and attributes, respectively, $w_{i,j}$ and $b_{i,j}$ are learned weight vectors and biases of the associated classifiers, $c$ indexes the $C = 6$ sections and $c_m$ indexes the $C_m$ values of the $m$-th attribute. Because not all attributes are present among all audio examples of a given machine type in the DCASE 2022 Task 2 dataset, the attribute loss in (3) is combined over all attributes in a multi-task learning fashion from the same embedding $z_A$, rather than learning disentangled feature dimensions for each attribute. If an attribute is unknown for an audio example, the corresponding term in the sum of (3) is ignored.

We note that our formulation of attribute learning in (3) as a multi-task learning problem with a different objective for each attribute differs from [21] where every possible combination of section and attribute corresponded to a different class.

### 2.2. Inference Approaches

The nearest neighbor (NN) algorithm is a simple and effective approach for anomaly detection [22, 23] given feature vectors of normal samples. As illustrated in Fig. 1, during inference we use the NN distance between a test embedding $z_q$ and all corresponding training set embeddings $z_q^{(j)}$ for computing an anomaly score, i.e.,

$$D_{\text{NN}}(z_q, \mathcal{D}) = \min_{j \in \mathcal{D}} D_{\cos}(z_q, z_q^{(j)}), \tag{4}$$

where $D_{\cos}(\cdot, \cdot)$ is the cosine distance between two embedding vectors. The disentangled model allows us to explore multiple inference approaches depending on which embedding dimensions we use for $z_q$ in (4) as discussed below.

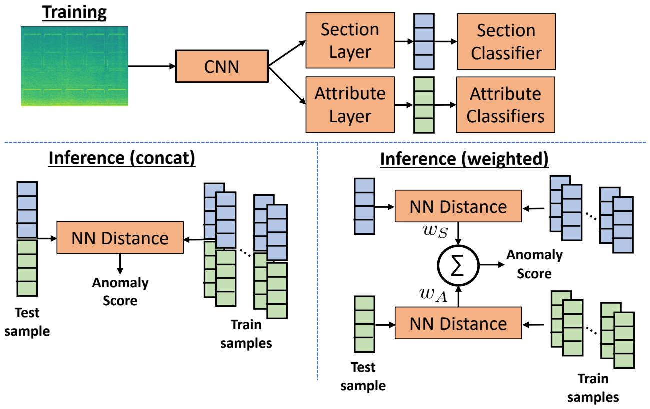

Figure 1: Block diagram of disentangled anomaly detector. In the figure, NN stands for Nearest Neighbor. In the training phase, exclusive latent spaces were assigned to sections and attributes.

**Disentangled Concatenated:** Use the concatenated embedding $z_C = [z_S^T, z_A^T]^T$ in (4) as shown in the bottom-left of Fig. 1.
**Disentangled Weighted:** As illustrated in the bottom-right of Fig. 1, we take a weighted average of NN distances separately computed for section embedding $z_S$ and attribute embedding $z_A$, i.e.,

$$D_{\text{NN}}^{\text{wt}}(z_S, z_A, \mathcal{D}) = w_S D_{\text{NN}}(z_S, \mathcal{D}) + w_A D_{\text{NN}}(z_A, \mathcal{D}) \tag{5}$$

where $w_S$ and $w_A$ are scalar weights, which are optimized after training is complete based on dev set performance. The best weights for each machine are shown in Table 4.
**Disentangled Sections:** Use only section embedding $z_S$ in (4).
**Disentangled Attributes:** Use only attribute embedding $z_A$ in (4).

At test time, the section label of the test sample is known, therefore, we limit the training set samples from $\mathcal{D}$ when computing the NN distance to be only those samples belonging to the appropriate section. Furthermore, our CNN architecture, detailed in Section 3.3, operates on spectrogram chunks of $T = 32$ time frames ($\sim$1 s), while each test sample is 10 s long. Using a chunk hop size of one frame, we obtain 282 embedding vectors per 10 s audio file. Following [22], we merged the embedding vectors for each sample by calculating their mean, except for valve where merging based on standard deviation provided significant gains. We then use the merged embedding vectors for computing the anomaly score.

## 3. EXPERIMENTAL SETUP

### 3.1. Dataset

There are seven different machine types in the DCASE 2022 Task 2 dataset [3] — ToyCar, ToyTrain, Bearing, Fan, Gearbox, Slider, and Valve. ToyCar and ToyTrain are from the ToyADMOS2 dataset [24], and the five other machines are from the MIMII DG dataset [25]. The data under each machine type is divided into sections, each of which corresponds to a specific type of domain shift. For example, in Fan, *section 00* refers to different machine noise between source and target domains, while *section 01* refers to different factory noise.

For each audio file, information about its section as well as one or more attributes is given. For machines belonging to the MIMII

DG dataset [3], only information on the domain shifting attribute, such as the type of machine noise in Fan's *section 00* and the type of factory noise in Fan's *section 01*, was present. For ToyCar and ToyTrain, which belong to the ToyADMOS2 dataset [24], information on all attributes was present in the filenames, even for those attributes that are not the domain shifting one. For the multi-task attribute learning (3), we make use of all present attributes, and represent them as categorical variables using all possible values found in the training set.

### 3.2. Audio Features and Training Strategy

The dataset contained 10 s audio files at a sampling rate of 16 kHz. We adopted short-time Fourier transform magnitude spectrograms as features for the neural network. The hop size was set to 32 ms and the window size was 128 ms (2048 samples). While training the neural network, the number of time steps for each audio example was 32 frames. Therefore, the input shape for the network was $1025 \times 32$. One epoch is defined as training the network on all 6000 audio files (six sections with 1000 examples in each section). For each audio file, a random chunk of 32 frames is selected for training. The advantages of this technique were reduced RAM usage, less chance of overfitting within epochs, and improved generalization compared with the baseline.

We adopted the Adam optimizer using a batch size of 32. In most cases, the learning rate was set to $10^{-4}$. For ToyCar, we found a minor improvement by setting it to $10^{-5}$. We trained the models for a maximum of 300 epochs, saved the model's weights every 5 epochs, tested the anomaly detector's performance on the development set, and selected the best performing model for each machine. We were unable to observe a clear relationship between the performances on the surrogate task and detection of anomalies. For instance, an improvement in the classification accuracy of sections (the surrogate task) was not necessarily accompanied by an improvement in anomaly detection. A similar observation has been made by previous studies using autoencoder-based models [10].

### 3.3. Neural Network Architecture

Morita et al. [22] found that the MobileFaceNet architecture [26] performed better than MobileNetV2 [27] as a feature extractor. We observed a similar improvement in initial experiments, and hence adopted MobileFaceNet. The parameter settings for MobileFaceNet can be found in Table 1. The output of the global depth-wise convolution (GDC) layer is a 512-D embedding vector. This is connected to the linear embedding layers (i.e., 1x1 convolutions) $\mathrm{L}^{Sec}$ and $\mathrm{L}^{Att}$ defined in Section 2, and associated softmax classification layers. Additionally, we explore minor modifications to the embedding and softmax layers as explained in Section 3.5.

### 3.4. Evaluation Metrics

We evaluate our models independently for each section and machine type using the three official metrics [3]: area under the ROC curve in the source (AUC (S)) and target (AUC (T)) domains, where the normal test samples are compared against anomalies from both domains, along with the domain agnostic partial AUC (pAUC) computed under low false-alarm-rate conditions.

For threshold-dependent metrics, we followed a similar approach to the baseline [3] and assumed the scores follow a gamma distribution. The parameters of the gamma distribution are estimated from the NN anomaly scores computed on the training set samples independently for each section (excluding self neighbors). For five machines, we set the anomaly detection threshold as the 90th percentile of the gamma distribution. For Fan and Bearing, we observed low sensitivity and hence adopted 60th percentile.

Table 1: MobileFaceNet [26] architecture, where all convolutions are 2D and dw-Conv refers to depth-wise convolution. In the network, Linear Conv $1 \times 1$ (sec) is connected to Softmax (sec), and Linear Conv $1 \times 1$ (att) is connected to the other softmax layers for attributes. For each layer, we show the expansion factor (**t**), number of channels (**c**), number of repeats (**n**), and stride (**s**). All convolutions excluding the final linear layers use PReLU activations.

| Input | Operator | t | c | n | s |
|---|---|---|---|---|---|
| 1x32x1025 | Conv 3x3 | - | 64 | 1 | 2 |
| 64x16x513 | dw-Conv 3x3 | - | 64 | 1 | 1 |
| 64x16x513 | Bottleneck | 2 | 64 | 5 | 2 |
| 64x8x257 | Bottleneck | 4 | 128 | 1 | 2 |
| 128x4x129 | Bottleneck | 2 | 128 | 6 | 2 |
| 128x2x65 | Bottleneck | 4 | 128 | 1 | 2 |
| 128x1x33 | Bottleneck | 2 | 128 | 2 | 1 |
| 128x1x33 | Conv 1x1 | - | 512 | 1 | 1 |
| 512x1x33 | Linear GDC 1x33 | - | 512 | 1 | 1 |
| 512x1x1 | Linear Conv 1x1 (sec) | - | 128 | 1 | 1 |
| 512x1x1 | Linear Conv 1x1 (att) | - | 128 | 1 | 1 |
| 128x1x1 | Softmax (sec) | - | 6 | - | - |
| 128x1x1 | Softmax (att$_1$) | - | $C_1$ | - | - |
| ⋮ | ⋮ | | ⋮ | | |
| 128x1x1 | Softmax (att$_M$) | - | $C_M$ | - | - |

### 3.5. Other Models Considered

**ArcFace** [29] was shown to improve class separability by adding angular margin to the loss. We investigated this technique's advantage by training on section indices. The feature scale and margin parameters were set to 32 and 0.5 respectively. We found ArcFace did not work well in a multi-task learning setting, probably because all attributes were not present in every example.

**Conventional Multi-task Learning (MTL)**: In this framework, the GDC layer from Table 1 is connected to a single 2D convolutional $1 \times 1$ layer with 256 channels. In other words, the features are in an entangled latent space.

**Machine-Specific Loss (MSL)**: Although not the focus of this paper, we also compare performance against the other two systems we submitted to the challenge: (S1) MSL as described in [20] and (S2) an ensemble of MSL and the attentive neural process (ANP) approach in [10] as detailed in Table 4.

## 4. RESULTS

**Development Set** results are shown in Table 2. Training using only section labels obtains an overall harmonic mean of 72.82%, which is significantly higher than both the baselines. This improvement is attributed to adopting the Nearest Neighbor algorithm during post-processing [22] and to our new training strategy explained in Section 3.2. Adopting ArcFace, which is essentially training on section indices with additive angular margin, improved the overall performance to 73.62%, while the AUC(T) improved from 67.34% to 71.79%. MTL, which trains on sections and attributes, obtained a lower overall performance of 72.47%, but improved the AUC(T) to 70.72%. Note that the MTL model does not use ArcFace. The Disentangled Sections (Disent_Sec) model only considers the section

Table 2: Results of different models on the development test set. We merge the three metrics and all sections to obtain a single number per machine using the harmonic mean. We also report the harmonic mean across machines and sections for each of the three metrics.

| **System** | **ToyCar** | **ToyTrain** | **Bearing** | **Fan** | **Gearbox** | **Slider** | **Valve** | **AUC (S)** | **AUC (T)** | **pAUC** | **Overall** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MSL+ANP (S2) | 76.43 | **59.96** | **73.93** | **68.89** | **85.37** | **85.93** | **95.83** | **87.55** | **73.43** | **70.36** | **76.43** |
| MSL (S1) | 76.43 | 59.37 | 73.93 | 68.85 | 83.03 | 85.37 | 95.63 | 86.78 | 73.34 | 69.68 | 75.93 |
| Disent_Wt (S4) | **76.95** | 59.74 | 72.07 | 63.91 | 81.38 | 85.14 | 94.50 | 86.09 | 71.65 | 68.21 | 74.57 |
| Disent_Cat (S3) | 76.43 | 58.67 | 67.09 | 63.18 | 80.99 | 85.37 | 95.01 | 84.61 | 70.59 | 67.02 | 73.34 |
| Disent_Sec | 76.84 | 56.64 | 72.07 | 62.35 | 81.04 | 84.84 | 94.42 | 86.43 | 68.64 | 68.01 | 73.45 |
| Disent_Att | 75.26 | 59.74 | 60.82 | 63.02 | 78.86 | 78.88 | 92.72 | 82.12 | 69.31 | 64.09 | 71.08 |
| MTL | 75.61 | 59.37 | 68.24 | 59.14 | 80.63 | 83.51 | 94.49 | 81.35 | 70.72 | 66.83 | 72.47 |
| Sec_ArcFace | 72.31 | 58.09 | 71.30 | 68.85 | 79.37 | 82.50 | 92.87 | 86.50 | 71.19 | 66.04 | 73.62 |
| Sec_Softmax | 76.20 | 52.85 | **73.93** | 64.39 | 81.43 | 85.89 | 90.11 | 86.05 | 67.34 | 67.92 | 72.82 |
| ANP-Boot | 59.84 | 50.87 | 55.54 | 55.31 | 64.38 | 64.11 | 52.63 | 69.26 | 50.87 | 54.24 | 57.10 |
| AE Baseline | 51.06 | 39.61 | 54.80 | 58.54 | 63.07 | 57.99 | 50.59 | 68.74 | 41.91 | 53.76 | 52.62 |
| MN Baseline | 54.23 | 51.18 | 59.16 | 57.21 | 59.91 | 50.26 | 62.42 | 63.87 | 50.14 | 55.69 | 56.01 |

Table 3: Official results of different models on the evaluation test set. We were unable to present the results in the same format as Table 2 because we do not have access to all the scores.

| System | ToyCar | | ToyTrain | | Fan | | Gearbox | | Bearing | | Slider | | Valve | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | pAUC | AUC | pAUC | AUC | pAUC | AUC | pAUC | AUC | pAUC | AUC | pAUC | AUC | pAUC | |
| Top rank [28] | 88.45 | **81.83** | **70.46** | **61.14** | **57.34** | **57.33** | 86.04 | 64.22 | 68.85 | 54.45 | **78.26** | **66.39** | **83.87** | **75.22** | **70.97** |
| Disent_Cat (S3) | **93.88** | 78.67 | 58.23 | 54.73 | 48.17 | 50.34 | **86.76** | **79.43** | **72.54** | 61.86 | 73.64 | 60.70 | 83.72 | 62.93 | 67.57 |
| Disent_Wt (S4) | 93.30 | 75.47 | 57.30 | 54.93 | 46.93 | 50.33 | 86.34 | 78.47 | 71.96 | **64.26** | 75.94 | 64.29 | 83.05 | 64.01 | 67.49 |
| MSL (S1) | **93.88** | 78.67 | 55.53 | 54.33 | 44.50 | 50.84 | 86.47 | 68.54 | 69.94 | 61.64 | 73.64 | 60.70 | 78.51 | 66.08 | 65.66 |
| MSL+ANP (S2) | **93.88** | 78.67 | 54.92 | 54.22 | 44.29 | 50.97 | 82.37 | 70.76 | 69.94 | 61.64 | 75.96 | 62.40 | 77.69 | 65.39 | 65.57 |
| MN Baseline | 42.79 | 53.44 | 51.22 | 50.98 | 50.34 | 55.22 | 51.34 | 48.49 | 58.23 | 52.16 | 62.42 | 53.07 | 72.77 | 65.16 | 54.02 |
| AE Baseline | 61.18 | 60.21 | 43.14 | 49.36 | 41.16 | 50.12 | 61.92 | 51.95 | 59.93 | 53.95 | 58.95 | 54.16 | 54.26 | 51.30 | 52.94 |

Table 4: Detailed model setups. We indicate the best MSL for S1. The ensemble weights (Ens. wt.) of S2 and the disentanglement weights (Disent. wt.) of S4 were calculated via a grid search.

| | | **Ens. wt. (S2)** | | **Disent. wt. (S4)** | |
|---|---|---|---|---|---|
| **Machine** | **MSL (S1)** | **MSL** | **ANP** | $w_S$ | $w_A$ |
| ToyCar | Disent_Cat | 0.60 | 0.40 | 0.90 | 0.10 |
| ToyTrain | MTL | 0.70 | 0.30 | 0.00 | 1.00 |
| Bearing | Sections only | 1.00 | 0.00 | 1.00 | 0.00 |
| Fan | ArcFace | 0.95 | 0.05 | 0.15 | 0.85 |
| Gearbox | Adversarial | 0.65 | 0.35 | 0.80 | 0.20 |
| Slider | Disent_Cat | 0.70 | 0.30 | 0.90 | 0.10 |
| Valve | Disent_Split | 0.80 | 0.20 | 0.90 | 0.10 |

embeddings during inference and obtains an overall performance of 73.45%. Although the overall performance is lower than that of ArcFace, it showed improvements for all machines except Bearing and Fan. The Disentangled Weighted (Disent_Wt) model obtains the highest overall performance for a single model without ensembling and machine-specific losses. The MSL and MSL+ANP systems showed improvements on the development set, but, as discussed next, worsened performance on the evaluation set.

**Evaluation Set** results on the DCASE challenge blind test set are shown in Table 3. Interestingly, both of our disentanglement models performed better than MSL and MSL+ANP. As indicated in Table 4, the MSL system used multi-task learning for ToyTrain, only sections for Bearing, ArcFace for Fan, adversarial training for Gearbox, and a slightly modified version of disentanglement for valve, where every attribute was assigned a separate embedding space. In all cases, our disentangled models outperformed the MSL systems. These observations convey that disentanglement is an effective technique for domain generalization.

However, optimizing disentanglement weights (Disent_Wt) on the dev set did not lead to improved performance over the simple concatenation approach (Disent_Cat). One hypothesis is that the optimized weights turned out to be slightly dataset-specific, which hurt generalization performance.

We ranked 5th out of 32 teams in the competition, obtaining an overall harmonic mean of 67.57%. We surpassed the baseline by 13.5% and trailed the top rank [28] by 3.4%. We surpassed the top rank for Bearing, Gearbox, and ToyCar(AUC). We believe that [28] adopted a better pipeline to train the feature extractor — (1) they average model weights from multiple epochs (2) they pre-train on all 7 machines and fine-tune for each machine. We hypothesize that our disentangled model can also be improved by incorporating these optimizations in the training pipeline.

## 5. CONCLUSION

In this study, we presented a disentangled multi-task learning framework for improved domain generalization in anomalous sound detection. We demonstrated that the disentangled model performs better than simple multi-task learning, or only learning based on domain-shared features (e.g., section indices). We also showed that there is increased flexibility in post-processing due to the multiple disentangled embedding spaces. In addition to the NN algorithm, we plan to explore other anomaly detection backends in the future, and to thoroughly evaluate the explainability of our approach.

## 6. REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, Nov. 2020, pp. 81–85.

[2] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions," in *Proc. DCASE*, Nov. 2021, pp. 186–190.

[3] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," *In arXiv e-prints: 2206.05876*, 2022.

[4] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *Proc. ICASSP*, Apr. 2015.

[5] E. Cakır and T. Virtanen, "Convolutional recurrent neural networks for rare sound event detection," in *Proc. DCASE*, Nov. 2017.

[6] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 1, pp. 212–224, 2018.

[7] T. Hayashi, T. Komatsu, R. Kondo, T. Toda, and K. Takeda, "Anomalous sound event detection based on wavenet," in *Proc. EUSIPCO*, Sept. 2018, pp. 2494–2498.

[8] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *Proc. ICASSP*, May 2020, pp. 271–275.

[9] R. Giri, F. Cheng, K. Helwani, S. V. Tenneti, U. Isik, and A. Krishnaswamy, "Group masked autoencoder based density estimator for audio anomaly detection," in *Proc. DCASE*, Nov. 2020, pp. 51–55.

[10] G. Wichern, A. Chakrabarty, Z.-Q. Wang, and J. Le Roux, "Anomalous sound detection using attentive neural processes," in *Proc. WASPAA*, 2021, pp. 186–190.

[11] P. Primus, V. Haunschmid, P. Praher, and G. Widmer, "Anomalous sound detection as a simple binary classification problem with careful selection of proxy outlier examples," in *Proc. DCASE*, Nov. 2020, pp. 170–174.

[12] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Proc. DCASE*, Nov. 2020, pp. 46–50.

[13] J. A. Lopez, H. Lu, P. Lopez-Meyer, L. Nachman, G. Stemmer, and J. Huang, "A speaker recognition approach to anomaly detection," in *Proc. DCASE*, Nov. 2020, pp. 96–99.

[14] K. Wilkinghoff, "Sub-cluster adacos: Learning representations for anomalous sound detection," in *Proc. IJCNN*, 2021, pp. 1–8.

[15] T. Inoue, P. Vinayavekhin, S. Morikuni, S. Wang, T. H. Trong, D. Wood, M. Tatsubori, and R. Tachibana, "Detection of anomalous sounds for machine condition monitoring using classification confidence," in *Proc. DCASE*, Nov. 2020, pp. 66–70.

[16] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu, "Generalizing to unseen domains: A survey on domain generalization," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[17] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 830–838.

[18] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Metric learning vs classification for disentangled music representation learning," in *Proc. ISMIR*, 2020.

[19] Z. Ding and Y. Fu, "Deep domain generalization with structured low-rank constraint," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 304–313, 2017.

[20] S. Venkatesh, G. Wichern, A. Subramanian, and J. Le Roux, "Disentangled surrogate task learning for improved domain generalization in unsupervised anomalous sound detection," DCASE2022 Challenge, Tech. Rep., July 2022.

[21] K. Wilkinghoff, "Utilizing sub-cluster adacos for anomalous sound detection under domain shifted conditions," DCASE2021 Challenge, Tech. Rep., July 2021.

[22] K. Morita, T. Yano, and K. Tran, "Anomalous sound detection using cnn-based features by self supervised learning," DCASE2021 Challenge, Tech. Rep., July 2021.

[23] M. Jones, D. Nikovski, M. Imamura, and T. Hirata, "Exemplar learning for extremely efficient anomaly detection in real-valued time series," *Data mining and knowledge discovery*, vol. 30, no. 6, pp. 1427–1454, 2016.

[24] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," *arXiv preprint arXiv:2106.02369*, 2021.

[25] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "Mimii dg: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," *arXiv preprint arXiv:2205.13879*, 2022.

[26] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Chinese Conference on Biometric Recognition*. Springer, 2018, pp. 428–438.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[28] Y. Zeng, H. Liu, L. Xu, Y. Zhou, and L. Gan, "Robust anomaly sound detection framework for machine condition monitoring," DCASE2022 Challenge, Tech. Rep., July 2022.

[29] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.