Why Do Angular Margin Losses Work Well for Semi-Supervised Anomalous Sound Detection?

Kevin Wilkinghoff, *Student Member, IEEE*, and Frank Kurth, *Senior Member, IEEE*

***Abstract*—State-of-the-art anomalous sound detection systems often utilize angular margin losses to learn suitable representations of acoustic data using an auxiliary task, which usually is a supervised or self-supervised classification task. The underlying idea is that, in order to solve this auxiliary task, specific information about normal data needs to be captured in the learned representations and that this information is also sufficient to differentiate between normal and anomalous samples. Especially in noisy conditions, discriminative models based on angular margin losses tend to significantly outperform systems based on generative or one-class models. The goal of this work is to investigate why using angular margin losses with auxiliary tasks works well for detecting anomalous sounds. To this end, it is shown, both theoretically and experimentally, that minimizing angular margin losses also minimizes compactness loss while inherently preventing learning trivial solutions. Furthermore, multiple experiments are conducted to show that using a related classification task as an auxiliary task teaches the model to learn representations suitable for detecting anomalous sounds in noisy conditions. Among these experiments are performance evaluations, visualizing the embedding space with t-SNE and visualizing the input representations with respect to the anomaly score using randomized input sampling for explanation.**

***Index Terms*—Anomaly detection, angular margin loss, compactness loss, domain generalization, explainable artificial intelligence, machine listening, representation learning.**

I. Introduction

SEMI-SUPERVISED anomalous sound detection (ASD) is the task of reliably detecting anomalous sounds while only having access to normal sounds for training a model [1]. Since anomalies occur only rarely by definition and usually are very diverse, collecting realistic anomalous samples for training a system is much more difficult and thus more costly than collecting normal data. Hence, a semi-supervised ASD setting is more realistic than a supervised ASD setting, for which anomalous sounds are available for training, because it substantially simplifies the data collection process. There are also unsupervised ASD settings, for which the training dataset may also contain anomalous samples and it is unknown whether a training sample is normal or anomalous. But for many applications, it can be ensured thaonly normal samples are collected for training and thus a semi-supervised setting can be assumed.

ASD has many applications. Examples are machine condition monitoring [2], [3], [4], medical diagnosis [5], [6], bioacoustic monitoring [7], [8], intrusion detection in smart home environments [9] and detecting crimes [10], [11] or accidents [12], [13]. Furthermore, detecting anomalous samples can also be understood as a subtask in acoustic open-set classification [14], [15], [16]. Throughout this work, we will use machine condition monitoring in domain-shifted conditions as an application example [4]. Here, the audio signals may contain one or several of the following three components: 1) normal machine sounds, 2) anomalous machine sounds and 3) background noise consisting of a mixture of many other sound events. The major difficulty of this ASD application is that anomalous components of machine sounds can be very subtle when being compared to the background noise making it difficult to reliably detect anomalous signal components. Furthermore, machine sounds and background noise can change substantially for different domain shifts, which we define as alterations in the (acoustic) environment or changes in parameter settings of the machines. The ASD system still needs to only detect anomalous signal components without frequently raising false alarms caused by any domain shift.

There are several strategies to train an ASD system for machine condition monitoring using only normal data. Among these strategies are generative models such as autoencoders [17], [18], [19], [20], [21], [22] or normalizing flows [23], [24] that directly try to model the probablity distribution of normal data, which is also called inlier modeling (IM) [3]. Another strategy is to use an auxiliary task, usually a classification task, for training a model to learn meaningful representations of the data (embeddings) that can be used to identify anomalies. Possible auxiliary tasks for machine condition monitoring are classifying between machine types [25], [26], [27], [28], [29] or, additionally, between different machine states and noise settings [30], [31], [32], [33], recognizing augmented and not augmented versions of normal data (self-supervised learning) [25] or predicting the activity of machines [32]. Using an auxiliary task to learn embeddings is also called outlier exposure (OE) [34] because normal samples belonging to other classes than a target class can be considered as proxy outliers [35]. Often an angular margin loss such as SphereFace [36], CosFace [37] or ArcFace [38] is utilized for training an OE model. Systems based on embeddings pre-trained on very large datasets [39], [40], [41] can be used, too. However, it has been shown that directly training a system on the data yields better ASD results, even when only very limited training data is available [42]. In addition, different strategies

Manuscript received 27 March 2023; revised 19 September 2023 and 20 November 2023; accepted 23 November 2023. Date of publication 28 November 2023; date of current version 8 December 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Romain Serizel. *(Corresponding author: Kevin Wilkinghoff.)*

The authors are with the Fraunhofer FKIE, 53343 Wachtberg, Germany (e-mail: kevin.wilkinghoff@ieee.org; frank.kurth@fkie.fraunhofer.de).

Digital Object Identifier 10.1109/TASLP.2023.3337153

can be combined by using an ensemble of multiple models [43], [44], [45].

Different strategies to train an ASD system have different strengths and weaknesses. Using an auxiliary task for training relies on additional meta-information to generate labels for a classification task whereas IM-based models do not need any labels. Furthermore, autoencoders can localize anomalies in the input space by visualizing an element-wise reconstruction error as done in [19], [21]. However, training ASD models by using an auxiliary task usually enhances their performance [46]. Even for IM-based models, performance can be significantly improved when utilizing meta information such as machine types. In [21] a class-conditioned autoencoder is used, in [44] not only spectral features but also the machine ID is encoded and decoded, and in [23] a normalizing flow is trained to assign lower likelihood to sounds of other machines and a higher likelihood to sounds of the target machine. As suspected in [32], [33], the most likely reason for the difference in performance is that, as stated before, recordings for machine condition monitoring are very noisy because of factory background noise. This is a problem for IM-based models because they cannot tell the difference between arbitrary sound events not emitted by a monitored machine and normal or anomalous sounds emitted by the machine. Both are considered equally important by the model. Moreover, anomalies present in these noisy audio recordings are usually very subtle when being compared to the noise or other sound events present in a recording making it even more difficult to detect potential anomalies. When being trained with an auxiliary task, a model learns to ignore noise, which can be assumed to be similar for all considered classes, and therefore to isolate the target machine sound by ignoring the uninformative background sound events. As a result, these models are more sensitive to changes of the machine sounds and have better anomaly detection capabilities.

Localizing and visualizing frequencies or temporal regions of recordings that are being considered anomalous is important for practical applications because users can better understand the decisions of the ASD system (explainable artificial intelligence (xAI) [47]). Furthermore, this may help to find the cause of mechanical failure and thus can simplify the maintenance process. As stated before, autoencoders can easily localize anomalies by using an element-wise reconstruction error. Additional investigations on visualizing and explaining ASD decisions include showing that decisions of ASD systems for machine condition monitoring largely rely on high-frequency information [48]. This has been visualized using local interpretable model-agnostic explanations (LIME) [49] applied to sounds (SLIME) [50]. Furthermore, uniform manifold approximation and projection (UMAP) [51] has been used to visualize representations of the data such as stacked consecutive frames of log magnitude spectrograms, log-mel magnitude spectrograms, or openL3 embeddings [46].

The goal of this work is to explain why angular margin losses work well for anomalous sound detection. To achieve this goal, the following contributions are made: First and foremost, it is theoretically proven that, after normalizing the embedding space, training an ASD model by minimizing an angular margin loss using an auxiliary task can be considered as minimizing a regularized one-class loss while being less affected by noise or non-target sound events present in the data. Moreover, it is experimentally verified that using an angular margin loss for training a model to discriminate between classes of an auxiliary task also leads to better ASD performance and thus is a better choice for an ASD task than minimizing a one-class loss such as an intra-class (IC) compactness loss with a single or multiple classes. Last but not least, a procedure for visualizing normal and anomalous regions of the input representations based on randomized input sampling for explanation (RISE) is presented. Using these visualizations, it is shown that normal and anomalous sounds cannot be distinguished from the highly complex background noise when training with a one-class loss. In contrast, when using an auxiliary task with multiple classes the model learns to ignore noise and isolate the targeted machine sound for monitoring their condition.

The paper is structured as follows: In Section II, various one-class losses and angular margin losses are reviewed. Section III presents our main theoretical results about the relation between these loss functions. Section IV contains a description of the experimental setup and all experimental evaluations consisting of performance evaluations, a comparison between losses during training, visualizing normal and anomalous regions of input representations as perceived by the system and visualizing the resulting embedding spaces. Section V consists of the conclusions of this work.

II. Loss Functions

In this section, a unified presentation and discussion of several loss functions that are needed for presenting one of the main results of this work in Section III will be given. The following notation will be used throughout the paper: X denotes the space of input data samples, $N \in \mathbb{N}$ the number of classes defined for an auxiliary task and $D \in \mathbb{N}$ the dimension of the embedding space.

A. One-Class Classification Losses

When training a model for ASD while only having access to normal data i.e. a single class, this is referred to as *one-class classification* and is some form of IM. The compactness loss [52], whose goal it is to project the data into a hypersphere of minimum volume, will serve as a representative of losses for one-class classification and is defined as follows.

Definition 1 (Compactness loss): Let $Y \subset X$ be finite. Let $\mathcal{P}$ denote the power set, Φ denote the space of network architectures for extracting embeddings and $W(\phi)$ denote the parameter space of $\phi \in \Phi$, i.e. $\phi : X \times W(\phi) \to \mathbb{R}^D$. Then, the *compactness loss* is defined as

$$\begin{aligned} &\mathcal{L}_{\text{comp}} : \mathcal{P}(X) \times \mathbb{R}^D \times \Phi \times W \to \mathbb{R}_+ \\ &\mathcal{L}_{\text{comp}}(Y, c, \phi, w) := \frac{1}{|Y|} \sum_{x \in Y} \|\phi(x, w) - c\|_2^2. \end{aligned} \quad (1)$$

The vector $c \in \mathbb{R}^D$ is called *center*.

After training, the (squared) Euclidean distance between the embedding of a given sample and the center can be utilized as an anomaly score: A greater distance indicates a higher likelihood for the sample to be anomalous. A *trivial solution* for minimizing the compactness loss with center $c \in \mathbb{R}^D$ is a parameter setting $w_c \in W(\phi)$ such that ϕ is the constant function $\phi(x, w_c) = c$ for all $x \in X$. It is of utmost importance to prevent that the model to be trained is able to learn such a trivial solution. Otherwise it is impossible to differentiate between normal and anomalous samples.

There are several strategies to prevent a model from learning a trivial solution. First of all, it needs to be ensured that $c \neq c_0 \in \mathbb{R}^D$ where $c_0 = \phi(x, w_0)$ is defined as the output of the network obtained by setting the weight parameters of model ϕ to zero. This is because we have $\phi(x, w_0) = c_0$ for all $x \in X$ as long as the model uses only linear operators, e.g. dense or convolutional layers, and all activation functions have zero as a fixed point, which is the case for most commonly used activation functions. In [52], is has been shown that using bias terms, bounded activation functions or a trainable center all enable the model to learn a constant function when using an additive weight decay regularization term and thus must also be avoided.

Another possibility to avoid trivial solutions is to impose additional tasks, so-called *auxiliary tasks*, not directly related to the ASD problem while training. Autoencoders [53], which are trained to first encode and then decode the input again and have many interesting applications by themselves such as denoising data [54], can also be viewed as a way to regularize one-class models. Here, the encoder is the one-class model mapping the input to an embedding space. Learning a constant function is not a (trivial) solution for the task because all necessary information for being able to completely reconstruct the input needs to be encoded. However, noise including other sound sources present in the input audio data needs to be encoded as well because otherwise the input cannot be reconstructed. Therefore, the noise heavily influences the embeddings and thus the embeddings can also be considered noisy. Depending on the complexity of the noise, most information contained in the embeddings is only related to the noise and not to the target sound to be analyzed and thus detecting anomalies using an autoencoder may be difficult. Moreover, in [52] it has been shown that using compactness loss, even for clean datasets, outperforms commonly used autoencoder architectures when detecting anomalies.

A second choice of an auxiliary task to prevent the model from learning a constant function as a trivial solution is a classification task. Defining multiple classes through an auxiliary task inherently prevents learning a constant function as this would not be a (trivial) solution to the imposed classification problem. In [55], an additional *descriptiveness loss* is used whose goal is to reduce inter-class similarity between classes of an arbitrary, external multi-class dataset, which is only used to regularize the one-class classification task. This is done by minimizing the standard categorical cross-entropy (CCE) loss for classification on this additional dataset as an auxiliary task. For each of the two tasks, another version of the same network with identical structure and tied weights is used. During training, both losses are jointly minimized using a weighted sum ensuring that the so-called reference network associated with the compactness loss does not learn a constant function because this would prevent the secondary network to be able to classify correctly.

Remark: The original definition of the compactness loss [52] also includes an additional weight decay term. Such a weight decay term can be used to complement any loss function and does not prevent the model from learning trivial solutions as it is still possible that the model learns to map everything to the center. Furthermore, all theoretical results presented in this work are valid regardless of whether this specific weight decay term is included or not. The proof of the main theorem can easily be modified to including the same weight decay term because it is just an additional additive term. Therefore, we omitted this term in the theoretical investigations of this work for the sake of simplicity while still using it in our experiments. However, we did not notice any significant effect on the performance.

For the remainder of this work, we propose to normalize all representations in the embedding space $\mathbb{R}^D$, meaning that $\|c\|_2 = 1 = \|\phi(x, w)\|_2$ for all $x \in X, w \in W(\phi)$ and centers $c \in \mathbb{R}^D$. This can easily be achieved by dividing the embeddings by their corresponding Euclidean norms. A normalization of the embedding space essentially reduces the dimension by one as evident by using stereographic projection. But doing so does not degrade the ASD performance because the dimension of the embedding space usually is larger than it needs to be.

Normalizing the embedding space has several advantages. Most importantly, the initialization of the centers is substantially simplified. In high-dimensional vector spaces i.i.d. random elements are almost surely approximately orthogonal [56]. Hence, all class centers can be randomly initialized by sampling from a uniform random distribution as also done in [33] and a careful strategy for initializing the class centers is not needed. This does not cause any problems e.g. by accidentally using class centers that are very similar to each other in terms of cosine similarity whereas the corresponding acoustic classes are very dissimilar or vice versa. Moreover, normalizing the centers ensures that all centers are distributed equidistantly and sufficiently far away from zero to avoid learning a trivial solution. Last but not least, normalizing the embeddings may even prevent numerical issues while training similar to when using batch normalization [57].

B. Angular Margin Losses

We will review the definition of ArcFace [38] as a representative of angular margin losses.

Definition 2 (ArcFace): Let $Y \subset X$ be finite and $l_j(x) \in \{0, 1\}$ denote the jth component of the categorical class label function $l \in L$ where L denotes the space of all functions $l : X \to \{0, 1\}^N$ with $\sum_{j=1}^N l_j(x) = 1$ for all $x \in X$. Let $\mathcal{P}$ denote the power set, Φ denote the space of network architectures for extracting embeddings and $W(\phi)$ denote the parameter space of $\phi \in \Phi$, thus $\phi : X \times W(\phi) \to \mathbb{R}^D$. Let $\operatorname{smax} : \mathbb{R}^N \to [0, 1]^N$ denote the softmax function, i.e.

$$\operatorname{smax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^N \exp(x_j)}. \tag{2}$$

Then, the *ArcFace* loss is defined as

$$\mathcal{L}_{\text{ang}} : \mathcal{P}(X) \times \mathcal{P}(\mathbb{R}^D) \times \Phi$$

$$\begin{aligned}&\times W \times L \times \mathbb{R}_+ \times \left[0, \frac{\pi}{2}\right] \to \mathbb{R}_+\\ &\mathcal{L}_{\text{ang}}(Y, C, \phi, w, l, s, m)\\ := -&\frac{1}{|Y|} \sum_{x \in Y} \sum_{j=1}^{N} l_j(x) \log(\text{smax}(s \cdot \cos_{\text{mar}}(\phi(x, w), c_j, m)))\end{aligned} \tag{3}$$

where $|C| = N$ and, in this case,

$$\begin{aligned}&\text{smax}(s \cdot \cos_{\text{mar}}(\phi(x, w), c_i, m))\\ := &\frac{\exp(s \cdot \cos_{\text{mar}}(\phi(x, w), c_i, m))}{\sum_{j=1}^{N} \exp(s \cdot \cos_{\text{mar}}(\phi(x, w), c_j, m \cdot l_j(x))}\end{aligned} \tag{4}$$

with

$$\cos_{\text{mar}}(x, y, m) := \cos(\arccos(\cos(x, y)) + m) \tag{5}$$

for cosine similarity

$$\cos(x, y) := \frac{\langle x, y\rangle}{\|x\|_2 \|y\|_2} \in [-1, 1]. \tag{6}$$

The vectors $c_j \in \mathbb{R}^D$ are called *class centers*, $m \in [0, \frac{\pi}{2}]$ is called *margin* and $s \in \mathbb{R}_+$ is called *scale parameter*.

Remark: When using mixup [58] for data augmentation, the definition of the class label function needs to be generalized to $l : X \to [0, 1]^N$ with $\sum_{j=1}^{N} l_j(x) = 1$ for all $x \in X$. In the experimental evaluations of this work, mixup will be used when training a model as this improves the ASD performance [29]. Furthermore, the theoretical results presented in this work still hold when using mixup but in the proofs only binary labels will be used for the sake of simplicity.

In [59], it has been shown that the choice of both hyperparameters, the scale parameter s and the margin m, can have a significant impact on the resulting performance. Strongly varying the magnitude of one of the individual parameters has a similar effect on the sensitivity of the posterior probabilities with respect to the angles as varying the other parameter. Both a scale parameter that is too large and a margin that is too small lead to very high posterior probabilities for the target class, approximately equal to one, even for relatively large angles. Therefore, the loss function is insensitive to changing the angle. A scale parameter that is too small limits the maximum posterior probability of the target class that can be achieved. Similarly, a margin that is too large also leads to relatively small posterior probabilities. Thus, in both cases the model still tries to adapt its parameters even when the angles are already small, which hinders convergence. Due to the similar behavior of both parameters, a single appropriately chosen parameter is sufficient for controlling the posterior probabilities and it has even been shown that an adaptive scale parameter outperforms using two tuned but fixed parameters. Therefore, we will assume that s is adaptive as specified for the AdaCos loss in [59] and set $m = 0$, i.e. $\cos_{\text{mar}}(x, y, 0) = \cos(x, y)$ for the remainder of this work. Formally, the definition of the AdaCos loss is the following.

Definition 3 (AdaCos): Using the same notation as in Definition 2, let $Y^{(t)} \subset Y$ denote all samples belonging to a mini-batch of size $B \in \mathbb{N}$, i.e. $|Y^{(t)}| = B$. Let $\theta_{x,i} := \arccos(\cos(\phi(x, w), c_i)) \in [0, \pi]$ and the *dynamically adaptive scale parameter* $\tilde{s}^{(t)} \in \mathbb{R}_+$ at training step $t \in \mathbb{N}_0$ be set to

$$\tilde{s}^{(t)} := \begin{cases}\sqrt{2} \cdot \log(N - 1) & \text{if } t = 0\\ \frac{\log B_{\text{avg}}^{(t)}}{\cos\left(\min(\frac{\pi}{4}, \theta_{\text{med}}^{(t)})\right)} & \text{else}\end{cases} \tag{7}$$

where $\theta_{\text{med}}^{(t)} \in [0, \pi]$ denotes the median of all angles $\theta_{x,i(x)}$ with $x \in X^{(t)}$ and $i(x) \in \{1, \ldots, N\}$ such that $l_i(x) = 1$ and

$$B_{\text{avg}}^{(t)} := \frac{1}{B} \sum_{x \in Y^{(t)}} \sum_{\substack{j=1\\ l_j(x) \neq 1}}^{N} \exp\left(\tilde{s}^{(t-1)} \cdot \cos(\phi(x, w), c_j)\right) \tag{8}$$

is the sample-wise average over all summed logits belonging to the non-corresponding classes. Then, the *AdaCos* loss is defined as

$$\begin{aligned}&\mathcal{L}_{\text{ada}} : \mathcal{P}(X) \times \mathcal{P}(\mathbb{R}^D) \times \Phi \times W \times L \to \mathbb{R}_+\\ &\mathcal{L}_{\text{ada}}(Y, C, \phi, w, l) := \mathcal{L}_{\text{ang}}(Y, C, \phi, w, l, \tilde{s}, 0).\end{aligned} \tag{9}$$

Remark: When using mixup [58] for data augmentation, $\theta_{\text{med}}^{(t)} \in [0, \pi]$ needs to be replaced with the median of the mixed-up angles as specified in [29].

The AdaCos loss can also be extended to using multiple centers for each class, called sub-clusters, instead of a single one. The idea of using these sub-clusters is to allow the network to learn more complex distributions than a normal distribution for each class enabling the model to have a more differentiated view on the embeddings when using the cosine similarity as an anomaly score. This has been shown to improve the ASD performance [29] and thus helps to differentiate between normal and anomalous samples.

Definition 4 (Sub-cluster AdaCos): Using the same notation as in Definitions 2 and 3, let $C_j \in \mathcal{P}(\mathbb{R}^D)$ with $|C_j| = M$ denote all centers belonging to class $j \in \{1, \ldots, N\}$. Let the *dynamically adaptive scale parameter* $\hat{s}^{(t)} \in \mathbb{R}_+$ at training step $t \in \mathbb{N}_0$ be set to

$$\hat{s}^{(t)} := \begin{cases}\sqrt{2} \cdot \log(N \cdot M - 1) & \text{if } t = 0\\ \frac{f_{\max}^{(t)} + \log \hat{B}_{\text{avg}}^{(t)}}{\cos\left(\min(\frac{\pi}{4}, \theta_{\text{med}}^{(t)})\right)} & \text{else}\end{cases} \tag{10}$$

with

$$\hat{B}_{\text{avg}}^{(t)} := \frac{1}{B} \sum_{x \in Y^{(t)}} \sum_{j=1}^{N} \sum_{c \in C_j} \exp\left(\hat{s}^{(t-1)} \cos(\phi(x, w), c) - f_{\max}^{(t)}\right) \tag{11}$$

and

$$f_{\max}^{(t)} := \max_{x \in Y^{(t)}} \max_{j=1}^{N} \max_{c \in C_j} \hat{s}^{(t-1)} \cdot \cos(\phi(x, w), c). \tag{12}$$

Then, the *sub-cluster AdaCos* loss is defined as

$$\begin{aligned}&\mathcal{L}_{\text{sc-ada}} : \mathcal{P}(X) \times \mathcal{P}(\mathcal{P}(\mathbb{R}^D)) \times \Phi \times W \times L \to \mathbb{R}_+\\ &\mathcal{L}_{\text{sc-ada}}(Y, C, \phi, w, l)\end{aligned}$$

$$:= -\frac{1}{|Y|}\sum_{x\in Y}\sum_{j=1}^{N} l_j(x)\log(\operatorname{smax}(\hat{s}\cdot\cos(\phi(x,w),C_j))) \tag{13}$$

where $|C| = N$ and, in this case,

$$\begin{aligned}&\operatorname{smax}(\hat{s}\cdot\cos(\phi(x,w),C_j))\\ &:= \sum_{c_j\in C_j}\frac{\exp(\hat{s}\cdot\cos(\phi(x,w),c_j))}{\sum_{k=1}^{N}\sum_{c_k\in C_k}\exp(\hat{s}\cdot\cos(\phi(x,w),c_k))}\end{aligned} \tag{14}$$

Remark: As shown in [29], for the sub-cluster AdaCos loss as defined above mixup [58] needs to be used. Otherwise, the dynamically adaptive scale parameter $\hat{s}^{(t)}$ grows exponentially.

For the compactness loss, there is no benefit of using sub-clusters. The reason is that an optimal solution of this sub-cluster compactness loss would correspond to the mean of the sub-clusters or, in case all embeddings are normalized, to its projection onto the unit sphere. Hence, there would be a single global optimum and this sub-cluster compactness loss would behave as if only a single sub-cluster is used. For the sub-cluster AdaCos loss, the situation is completely different because the softmax function is applied to all individual sub-clusters and the sum over the resulting scores is taken. This makes the resulting softmax probability, and thus also the loss function, symmetric with respect to the corresponding sub-clusters of an individual class. Therefore, the loss is invariant to changing the position of an embedding on the hypersphere as long as the sum of the distances to the sub-clusters is the same. Hence, also the space of optimal solutions grows with respect to the number of sub-clusters. However, due to the dependence on the sub-clusters of the other classes caused by the softmax function, this invariance is a simplification and the real situation is more complex.

III. Relation Between One-Class Losses and Angular Margin Losses

For the proof of the main theoretical result of this work, the following basic identity is needed.

Lemma 5: For $x, y \in \mathbb{R}^D$ with $\|x\|_2 = \|y\|_2 = 1$, it holds that

$$\cos(x,y) = 1 - \frac{\|x-y\|_2^2}{2}. \tag{15}$$

Proof: See Appendix.

Remark: This lemma also shows that for normalized embeddings using Euclidean distance and using cosine distance, which in this case is equal to the standard scalar product, are equivalent for computing an anomaly score.

Now, the theorem itself follows.

Theorem 6: Let $Y_j := \{x \in Y : l_j(x) = 1\}$. Then minimizing $\mathcal{L}_{\text{sc-ada}}(Y, C, \phi, w, l)$ with gradient descent minimizes all IC compactness losses with weighted gradients given by

$$\begin{aligned}&\frac{\hat{s}}{2}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{x\in Y_i}\sum_{c_i\in C_i}P(\tau(\phi(x,w))=c_i|\tau(\phi(x,w))\in C_i)\\ &\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2\end{aligned} \tag{16}$$

while maximizing all inter-class compactness losses with weighted gradients given by

$$\begin{aligned}&-\frac{\hat{s}}{2}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{x\in Y_i}\sum_{k=1}^{N}\sum_{c_k\in C_k}P(\tau(\phi(x,w))=c_k)\\ &\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2\end{aligned} \tag{17}$$

where

$$\begin{aligned}&P(\tau(\phi(x,w))=c_i|\tau(\phi(x,w))\in C_i)\\ &:= \frac{\exp(\hat{s}\cdot\cos(\phi(x,w),c_i))}{\sum_{c_i'\in C_i}\exp(\hat{s}\cdot\cos(\phi(x,w),c_i'))}\end{aligned} \tag{18}$$

and

$$\begin{aligned}&P(\tau(\phi(x,w))=c_k)\\ &:= \frac{\exp(\hat{s}\cdot\cos(\phi(x,w),c_k))}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}\exp(\hat{s}\cdot\cos(\phi(x,w),c_k'))}\end{aligned} \tag{19}$$

with a cluster assignment function $\tau : \mathbb{R}^D \to \mathbb{R}^D$ given by

$$\tau(z,C) = \arg\max_{c\in C}\cos(z,c). \tag{20}$$

Proof: Let $x \in Y$, $\phi \in \Phi$ and $\hat{s} \in \mathbb{R}_+$ be fixed and $i \in \{1,\ldots,N\}$ such that $l_i(x) = 1$ and $l_j(x) = 0$ for $j \neq i$. To simplify notation, define $e(w,c) := \exp(\hat{s}\cdot\cos(\phi(x,w),c))$. Using Lemma 5, we see that

$$\begin{aligned}&\frac{\partial}{\partial w}\log\left(\sum_{c_i\in C_i}e(w,c_i)\right)\\ &= \frac{\sum_{c_i\in C_i}e(w,c_i)\cdot\hat{s}\cdot\frac{\partial}{\partial w}\cos(\phi(x,w),c_i))}{\sum_{c_i'\in C_i}e(w,c_i')}\\ &= -\frac{\hat{s}}{2}\sum_{c_i\in C_i}\frac{e(w,c_i)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2}{\sum_{c_i'\in C_i}e(w,c_i')}\end{aligned}$$

and similarly

$$\begin{aligned}&\frac{\partial}{\partial w}\log\left(\sum_{k=1}^{N}\sum_{c_k\in C_k}e(w,c_k)\right)\\ &= \frac{\sum_{k=1}^{N}\sum_{c_k\in C_k}e(w,c_k)\cdot\hat{s}\cdot\frac{\partial}{\partial w}\cos(\phi(x,w),c_k))}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\\ &= -\frac{\hat{s}}{2}\sum_{k=1}^{N}\sum_{c_k\in C_k}\frac{e(w,c_k)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\\ &= -\frac{\hat{s}}{2}\sum_{c_i\in C_i}\frac{e(w,c_i)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\\ &\quad-\frac{\hat{s}}{2}\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{c_k\in C_k}\frac{e(w,c_k)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}.\end{aligned}$$

Using both identities, we obtain

$$\begin{aligned}
&\frac{\partial}{\partial w}\sum_{j=1}^{N} l_j(x)\log(\text{smax}(\hat{s}\cdot\cos(\phi(x,w),C_j)))\\
&=\frac{\partial}{\partial w}\log\left(\sum_{c_i\in C_i}\frac{e(w,c_i)}{\sum_{k=1}^{N}\sum_{c_k\in C_k}e(w,c_k)}\right)\\
&=\frac{\partial}{\partial w}\log\left(\sum_{c_i\in C_i}e(w,c_i)\right)-\frac{\partial}{\partial w}\log\left(\sum_{k=1}^{N}\sum_{c_k\in C_k}e(w,c_k)\right)\\
&=-\frac{\hat{s}}{2}\sum_{c_i\in C_i}\frac{e(w,c_i)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2}{\sum_{c_i'\in C_i}e(w,c_i')}\\
&\quad+\frac{\hat{s}}{2}\sum_{c_i\in C_i}\frac{e(w,c_i)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\\
&\quad+\frac{\hat{s}}{2}\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{c_k\in C_k}\frac{e(w,c_k)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\\
&=-\frac{\hat{s}}{2}\left(\sum_{c_i\in C_i}e(w,c_i)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2\right.\\
&\quad\cdot\left(\frac{1}{\sum_{c_i'\in C_i}e(w,c_i')}-\frac{1}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\right)\\
&\quad\left.-\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{c_k\in C_k}\frac{e(w,c_k)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\right)\\
&=-\frac{\hat{s}}{2}\left(\sum_{c_i\in C_i}e(w,c_i)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2\right.\\
&\quad\cdot\left(\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{c_k\in C_k}\right.\\
&\quad\left.\frac{e(w,c_k)}{(\sum_{c_i'\in C_i}e(w,c_i'))(\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k'))}\right)\\
&\quad\left.-\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{c_k\in C_k}\frac{e(w,c_k)\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\right)\\
&=-\frac{\hat{s}}{2}\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{c_k\in C_k}\frac{e(w,c_k)}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\\
&\quad\cdot\left(\sum_{c_i\in C_i}\frac{e(w,c_i)}{\sum_{c_i'\in C_i}e(w,c_i')}\cdot\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2\right.\\
&\quad\left.-\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2\right)\\
&=-\frac{\hat{s}}{2}\sum_{k=1}^{N}\sum_{c_k\in C_k}\underbrace{\frac{e(w,c_k)}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}}_{=P(\tau(\phi(x,w))=c_k)}\\
&\quad\cdot\sum_{c_i\in C_i}\underbrace{\frac{e(w,c_i)}{\sum_{c_i'\in C_i}e(w,c_i')}}_{=P(\tau(\phi(x,w))=c_i|\tau(\phi(x,w))\in C_i)}\\
&\quad\cdot\left(\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2-\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2\right)
\end{aligned}$$

where we used that

$$\begin{aligned}
&\frac{1}{\sum_{c_i'\in C_i}e(w,c_i')}-\frac{1}{\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k')}\\
&=\frac{\sum_{k=1}^{N}\sum_{c_k\in C_k}e(w,c_k)-\sum_{c_i\in C_i}e(w,c_i)}{(\sum_{c_i'\in C_i}e(w,c_i'))(\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k'))}\\
&=\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{c_k\in C_k}\frac{e(w,c_k)}{(\sum_{c_i'\in C_i}e(w,c_i'))(\sum_{k=1}^{N}\sum_{c_k'\in C_k}e(w,c_k'))}.
\end{aligned}$$

Now, summing over all samples $x\in Y$, normalizing with $|Y|$ and taking the additive inverse yields the desired result.

When using mixup, the right hand side of the last equation needs to be replaced with a weighted sum of two terms, each corresponding to one of the two classes that are mixed-up, because there are $i_1,i_2\in\{1,\ldots,N\}$ such that $l_{i_1}(x)\neq 0\neq l_{i_2}(x)$. Otherwise, the proof is exactly the same. In conclusion, the proven result still holds for mixed-up samples but includes two similar terms instead of one term. □

Corollary 7: Minimizing $\mathcal{L}_{\text{ada}}(Y,C,\phi,w,l)$ with gradient descent is equivalent to minimizing

$$\begin{aligned}
&-\frac{\tilde{s}}{2}\sum_{k=1}^{N}\text{smax}(\hat{s}\cdot\cos(\phi(x,w),c_k))\\
&\cdot\left(\frac{\partial}{\partial w}\|\phi(x,w)-c_i\|_2^2-\frac{\partial}{\partial w}\|\phi(x,w)-c_k\|_2^2\right). \qquad (21)
\end{aligned}$$

Proof: The proof of Theorem 6 does not depend on the exact structure of the dynamically adaptive scale parameter and thus also holds for the standard AdaCos loss by replacing $\hat{s}$ with $\tilde{s}$ and using only a single sub-cluster for each class. □

This theorem shows that using an angular margin loss such as the AdaCos loss is essentially the same strategy as proposed in [55] and applied to ASD in [27], i.e. using a compactness loss for increasing IC similarity, as defined in Definition 1, and a so-called descriptiveness loss to decrease inter-class similarity. However, there are differences between both approaches. When minimizing an angular margin loss, inter-class compactness losses are used to decrease inter-class similarity instead of a standard CCE loss. Second, when using two loss functions one usually has to tune a weight parameter to create a weighted sum of both loss terms, which is not needed for an angular margin loss and impossible without access to anomalous samples. Furthermore, the gradients belonging to individual samples

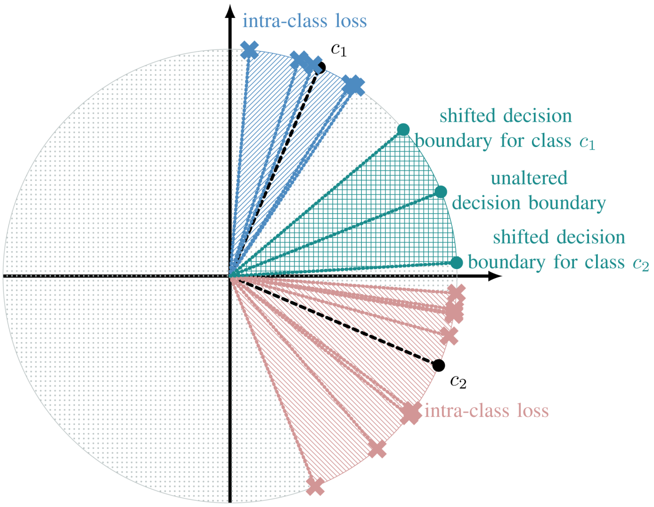

intra-class loss
c_1
shifted decision boundary for class c_1
unaltered decision boundary
shifted decision boundary for class c_2
c_2
intra-class loss

Fig. 1. Illustration of IC compactness losses and the angular margin to be ensured between the classes for $D = 2, N = 2, M = 1$. Intra-class losses are computed by summing all distances of samples to their corresponding class centers (blue and red areas). Inter-class losses are computed by summing all distances of samples to their corresponding decision boundaries. An unaltered decision boundary is exactly the midpoint between the class centers. When using an angular margin loss, the decision boundaries to the other classes are essentially shifted closer to the class center for which the inter-class loss is computed (see Fig. 1 in [60]). This explicitly ensures a margin between the classes, which is depicted by the green area.

are weighted with specific softmax probabilities giving more emphasis the closer the sub-clusters are. As these weights are non-uniform in general, this explicitly shows why using multiple sub-clusters is not equivalent to using a single sub-cluster given by the projection of the mean of the sub-clusters onto the hypersphere as it is the case for an IC compactness loss with multiple sub-clusters. Last but not least, an angular margin loss explicitly ensures a margin between classes, as illustrated in Fig. 1, whereas a combination of compactness losses and a CCE loss only implicitly does this by increasing intra-class similarity. Note that, in [55], inter-class similarity is decreased on another dataset using less relevant classes because only a single class is available on the target dataset. Because of these differences, directly minimizing an angular margin loss leads to a different solution than minimizing a combination of IC losses and a descriptiveness loss.

Note that the IC compactness loss with multiple classes can also be considered a prototypical loss [61] or angular prototypical loss [62] as used for few-shot classification [63], which defines settings where only very few training samples, called shots, are available for each class. The only difference between these prototypical losses and an angular margin loss is that, for prototypical losses, the center vectors are re-calculated as the means of embeddings belonging to corresponding classes by using a so-called support set during training while, for an angular margin loss, the class centers are fixed or adaptable parameters of the network. Hence, this theorem also shows that angular margin losses are a suitable choice for few-shot classification as shown for open-set sound event classification [42] and few-shot keyword spotting [64].

Choosing a classification task as an auxiliary task prevents learning a constant function as a trivial solution. The reason is that, for such a classification task, an optimal solution is a classifier that maps each sample to its corresponding class center and thus corresponds to jointly learning multiple trivial solutions, one for each class, instead of only learning a constant function. As long as each anomalous sample belongs to a well-defined normal class used during training, this optimal solution would yield representations not suitable for detecting anomalies as they would not be distinguishable from representations obtained with normal samples. However, obtaining such a perfect classifier is much more difficult than learning a constant mapping for a single class and thus training a single model to classify between multiple classes already prevents trivial solutions as long as the classification problem itself is not trivial e.g. by consisting of only a single class. Still, in [33] it has been shown that the ASD performance can be improved by applying the same three strategies as used for the compactness loss [52], namely 1) not using bias terms, 2) not using bounded activation functions and 3) not using trainable class centers. The most likely reason is that these strategies prevent the model to learn trivial solutions, leading to less informative embeddings, for individual classes that are easily recognized.

IV. Experimental Results

Using one-class losses and angular margin losses for ASD will now be compared experimentally.

A. Dataset

For most experiments conducted in this work, the DCASE2022 ASD dataset [4] of the task titled “Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques” has been used. The dataset consists of recordings of machine sounds with background factory noise. Each recording has a single channel, a length of ten seconds and a sampling rate of 16 kHz and belongs to one of the seven machine types “fan”, “gearbox”, “bearing”, “slide rail”, “valve” from MIMII DG [65] and “toy car”, “toy train” from ToyADMOS2 [66]. For each machine type, there are six different so-called sections each of which is dedicated to a specific type of domain shift. A domain shift means that the characteristics of a machine sound differ in some way between a source domain with many training samples and a target domain with only few training samples. These shifts can be caused by physical changes of the machines e.g. caused by replacing parts for maintenance, or changes in the acoustical environment e.g. a different background noise or using different recording devices. Ideally, the ASD system is able to reliably detect anomalies despite these domain shifts without the need for adapting the system (domain generalization [67]).

The dataset is divided into a development and an evaluation split each containing recordings of 21 sections, three for each machine type. For each recording, information about the machine type and section are given. For the training datasets, domain information (“source” or “target”) and additional attribute information such as states of machine types or noise conditions

TABLE I
STRUCTURE OF THE DCASE2022 ASD DATASET

subset	split	number of recordings (per section)			
		source domain		target domain	
		normal	anomalous	normal	anomalous
development	training	990	0	10	0
development	test	50	50	50	50
evaluation	training	990	0	10	0
evaluation	test	50	50	50	50

are given for each recording. For the test datasets, no domain information and no additional attribute information are given. The exact structure of the dataset can be found in Table I. The task of an ASD system is to reliably detect anomalous samples regardless of whether a sample belongs to a source or target domain, i.e. using a single decision threshold for both domains of a section.

Some of the experiments have also been conducted on the DCASE2023 ASD dataset [68], [69] belonging to the task "First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring". Similar to the DCASE2022 ASD dataset, this dataset is also aimed at domain generalization for ASD with the following differences. First and foremost, the development and evaluation split of the dataset contain different machine types. The development set contains the same machine types as the DCASE2022 dataset, namely "fan", "gearbox", "bearing", "slide rail", "valve" from MIMII DG [65] and "toy car", "toy train" from ToyADMOS2 [66]. The evaluation set contains seven completely different machine types, namely "toy drone", "n-scale toy train", "vacuum", and "toy tank" from [70] and "bandsaw", "grinder", "shaker" from [65]. Furthermore, for each machine type there is only a single section. This lowers the difficulty of the auxiliary classification task and thus makes it more difficult to extract embeddings, which are sensitive to anomalous changes of the target sounds.

For the DCASE ASD datasets, two performance measures are used to evaluate the performance of individual ASD systems. One metric is the area under the ROC curve (AUC), the other metric is the partial area under the ROC curve (pAUC) [71], which is the AUC calculated over a low false positive rate ranging from 0 to p with $p = 0.1$ in this case. The pAUC is used as an additional metric because decision thresholds for machine condition monitoring are usually set to a value that gives a low number of false alarms and thus this area of the receiver operating characteristic (ROC) curve is of particular interest. Both are threshold-independent metrics allowing a more objective comparison between different ASD systems than threshold-dependent metrics [1], [72].

B. System Description

The focus of this work is to explain why angular margin losses work well for ASD. This requires using different loss functions for training an ASD system. To this end, the conceptually simple state-of-the-art system presented in [33], which only consists of a single model and uses the same settings for all machine types, is utilized. For all experiments conducted in this work, only the loss function used for training the system is altered. The system utilizes a magnitude spectrogram as well as the whole magnitude spectrum as input representations and uses two different convolutional sub-models for handling these, resulting in two different embeddings. Then, both embeddings are concatenated to obtain a single embedding and the sub-cluster AdaCos loss [29] is applied with 16 sub-clusters, which are initialized uniformly at random, for training the model. For the magnitude spectrogram, temporal mean normalization is applied to reduce the effect of different acoustic domains and make both input feature representations a bit more different by removing constant frequency information from the spectrograms. Furthermore, the model does not use bias terms or trainable clusters as this improves the ASD performance by avoiding trivial solutions as discussed before. The model is trained for 10 epochs with a batch size of 64 using mixup [58] with a uniform distribution for sampling the mixing coefficient and is implemented in Tensorflow [73].

After training the model using an auxiliary classification task, embeddings are extracted for the recordings. For each section of the dataset, k-means with $k = 16$ is applied to all normal training samples belonging to the source domain of this section. The goal is to represent the distribution of the normal embeddings and be able to compute an anomaly score by taking the minimum cosine distance to the mean embeddings belonging to the same section as a given test sample. Note that these means do not correspond to the sub-clusters as some sub-clusters may not have been used by the network during training. It is possible that the embeddings are clustered between the sub-clusters due to the complex dependence between the sub-clusters of the other classes. Still, it has been shown taking the same number of clusters usually performs best [29]. Since there are only 10 normal samples available for the target domain, the minimum over the direct cosine distances to the corresponding embeddings is used. As a last step, the minimum of the minimum cosine distances belonging to both domains is used to have an ASD system that generalizes to both domains. Hence, a higher anomaly score indicates anomalous sounds whereas a smaller value indicates normal sounds. More details about the system including a hyperlink to an open-source implementation can be found in [33].

C. Performance Evaluations

Regardless of the loss function, training the ASD model without using anomalous samples is not directly targeting the ASD performance but only indirectly since the auxiliary task is aimed at obtaining embeddings suitable for ASD. Although, there is a strong relation between the auxiliary and the ASD task, as otherwise training an ASD model by using an auxiliary task would not lead to usable representations, the actual ASD performance needs to be evaluated experimentally and cannot be investigated theoretically because there are no anomalous samples available during training. Therefore, the resulting ASD performances obtained by minimizing both types of loss functions, angular margin losses and one-class losses, using individual auxiliary classification tasks will be evaluated first. Furthermore, a combined loss consisting of the sum of the mean of the IC compactness losses and an additional softmax layer with a CCE

TABLE II
ASD Performance Obtained With Different Losses Using Different Auxiliary Tasks

DCASE2022 development set							
loss	classes of auxiliary task (number of classes)	source domain		target domain		both domains	
		AUC	pAUC	AUC	pAUC	AUC	pAUC
IC compactness loss (Def. 1)	none (1)	56.4 ± 1.4	53.9 ± 0.6	53.6 ± 0.9	52.6 ± 0.3	55.1 ± 1.2	52.6 ± 0.4
IC compactness loss (Def. 1)	machine types (7)	66.5 ± 2.9	60.6 ± 0.6	63.6 ± 2.2	57.1 ± 0.9	65.0 ± 1.7	57.8 ± 0.6
IC compactness loss (Def. 1)	machine types and sections (42)	77.6 ± 1.7	70.5 ± 0.9	75.3 ± 0.9	63.3 ± 0.8	76.4 ± 0.9	63.5 ± 0.6
IC compactness loss (Def. 1)	machine types and sections, models trained individually (1)	50.0 ± 2.3	52.1 ± 0.6	51.7 ± 1.8	52.2 ± 0.4	51.8 ± 1.8	51.4 ± 0.4
IC compactness loss (Def. 1)	machine types, sections and attribute information (342)	80.7 ± 1.9	73.7 ± 1.0	74.5 ± 0.9	62.1 ± 1.2	78.1 ± 0.8	63.3 ± 0.9
IC compactness loss (Def. 1) + CCE	machine types, sections and attribute information (342)	82.5 ± 0.7	75.2 ± 0.7	75.5 ± 0.6	61.2 ± 1.6	79.0 ± 0.6	64.8 ± 0.9
AdaCos loss (Def. 3)	machine types, sections and attribute information (342)	83.0 ± 1.3	75.2 ± 1.8	75.4 ± 1.0	60.9 ± 0.8	79.2 ± 0.9	64.3 ± 0.7
sub-cluster AdaCos loss (Def. 4)	machine types, sections and attribute information (342)	$\mathbf{84.2 \pm 0.8}$	$\mathbf{76.5 \pm 0.9}$	$\mathbf{78.5 \pm 0.9}$	$\mathbf{62.5 \pm 0.9}$	$\mathbf{81.4 \pm 0.7}$	$\mathbf{66.6 \pm 0.9}$
DCASE2022 evaluation set							
loss	classes of auxiliary task (number of classes)	source domain		target domain		both domains	
		AUC	pAUC	AUC	pAUC	AUC	pAUC
IC compactness loss (Def. 1)	none (1)	49.9 ± 0.8	50.6 ± 0.4	51.0 ± 0.4	51.0 ± 0.7	50.9 ± 0.5	50.3 ± 0.4
IC compactness loss (Def. 1)	machine types (7)	59.6 ± 1.3	56.9 ± 0.5	57.6 ± 1.8	53.8 ± 0.9	59.3 ± 1.5	54.6 ± 0.6
IC compactness loss (Def. 1)	machine types and sections (42)	70.8 ± 1.2	62.1 ± 0.7	61.7 ± 0.8	55.4 ± 1.0	66.3 ± 0.6	56.5 ± 0.4
IC compactness loss (Def. 1)	machine types and sections, models trained individually (1)	52.9 ± 1.4	51.7 ± 0.5	54.5 ± 0.6	51.6 ± 0.3	54.2 ± 0.8	51.2 ± 0.3
IC compactness loss (Def. 1)	machine types, sections and attribute information (342)	73.7 ± 0.5	63.4 ± 0.7	67.9 ± 1.0	57.8 ± 1.3	70.9 ± 0.6	58.5 ± 0.9
IC compactness loss (Def. 1) + CCE	machine types, sections and attribute information (342)	74.7 ± 0.7	64.9 ± 1.1	69.2 ± 0.7	59.8 ± 1.3	71.9 ± 0.6	59.5 ± 1.0
AdaCos loss (Def. 3)	machine types, sections and attribute information (342)	76.3 ± 1.0	$\mathbf{66.0 \pm 0.5}$	$\mathbf{69.9 \pm 0.8}$	$\mathbf{59.9 \pm 1.5}$	73.2 ± 0.4	$\mathbf{60.1 \pm 0.9}$
sub-cluster AdaCos loss (Def. 4)	machine types, sections and attribute information (342)	$\mathbf{76.8 \pm 0.8}$	65.8 ± 0.2	69.8 ± 0.5	59.7 ± 1.1	$\mathbf{73.4 \pm 0.5}$	59.8 ± 0.8
DCASE2023 development set							
loss	classes of auxiliary task (number of classes)	source domain		target domain		both domains	
		AUC	pAUC	AUC	pAUC	AUC	pAUC
IC compactness loss (Def. 1)	none (1)	50.7 ± 3.5	52.6 ± 0.3	45.3 ± 1.9	50.1 ± 0.5	48.9 ± 1.4	50.9 ± 0.4
IC compactness loss (Def. 1)	machine types (14)	67.3 ± 2.7	63.0 ± 1.4	67.8 ± 1.2	$\mathbf{58.6 \pm 1.1}$	67.4 ± 1.4	$\mathbf{59.4 \pm 1.1}$
IC compactness loss (Def. 1)	machine types, models trained individually (1)	46.7 ± 1.9	51.7 ± 0.6	45.9 ± 3.2	50.4 ± 0.8	47.6 ± 2.1	50.7 ± 0.6
IC compactness loss (Def. 1)	machine types and attribute information (186)	67.6 ± 2.5	61.6 ± 1.2	70.0 ± 2.4	56.4 ± 1.9	68.3 ± 1.9	57.1 ± 1.3
IC compactness loss (Def. 1) + CCE	machine types and attribute information (186)	$\mathbf{70.1 \pm 1.5}$	$\mathbf{63.3 \pm 1.3}$	71.0 ± 1.3	55.5 ± 1.1	70.4 ± 1.0	56.7 ± 0.8
AdaCos loss (Def. 3)	machine types and attribute information (186)	69.8 ± 1.5	62.8 ± 1.3	72.1 ± 1.2	55.4 ± 1.7	$\mathbf{71.2 \pm 0.7}$	56.8 ± 1.2
sub-cluster AdaCos loss (Def. 4)	machine types and attribute information (186)	69.4 ± 1.5	61.4 ± 1.5	$\mathbf{72.4 \pm 1.6}$	55.3 ± 1.2	71.0 ± 1.2	56.3 ± 1.1
DCASE2023 evaluation set							
loss	classes of auxiliary task (number of classes)	source domain		target domain		both domains	
		AUC	pAUC	AUC	pAUC	AUC	pAUC
IC compactness loss (Def. 1)	none (1)	51.8 ± 2.1	51.4 ± 1.2	50.0 ± 1.9	50.5 ± 0.7	51.6 ± 0.9	50.8 ± 0.6
IC compactness loss (Def. 1)	machine types (14)	59.3 ± 1.9	54.4 ± 0.6	54.3 ± 2.1	51.2 ± 0.5	56.7 ± 1.2	52.0 ± 0.6
IC compactness loss (Def. 1)	machine types, models trained individually (1)	51.3 ± 0.7	51.9 ± 0.7	54.7 ± 1.7	52.3 ± 0.8	53.2 ± 1.0	51.5 ± 0.6
IC compactness loss (Def. 1)	machine types and attribute information (186)	$\mathbf{73.0 \pm 1.9}$	62.1 ± 1.4	58.9 ± 2.7	55.1 ± 1.4	64.1 ± 1.8	55.6 ± 0.8
IC compactness loss (Def. 1) + CCE	machine types and attribute information (186)	72.6 ± 1.4	$\mathbf{62.5 \pm 1.9}$	62.2 ± 2.6	56.2 ± 0.8	67.2 ± 0.7	$\mathbf{58.0 \pm 0.9}$
AdaCos loss (Def. 3)	machine types and attribute information (186)	72.3 ± 1.7	62.1 ± 1.4	61.6 ± 3.1	$\mathbf{56.4 \pm 1.1}$	67.0 ± 1.5	57.4 ± 0.9
sub-cluster AdaCos loss (Def. 4)	machine types and attribute information (186)	72.1 ± 1.9	61.3 ± 1.5	$\mathbf{62.3 \pm 2.7}$	56.0 ± 0.7	$\mathbf{67.3 \pm 1.3}$	57.4 ± 0.7

Harmonic means of all AUCs and PAUCs over all pre-defined sections of the dataset are depicted in percent. Arithmetic mean and standard deviation of the results over five independent trials are shown. Best results in each column are highlighted with bold letters.

loss for classification, as proposed in [55], is evaluated. The results can be found in Table II. Note that it is also possible to divide the classification task into several different classification tasks as for example one task for the machine type and other ones for all or specific attributes [30], [31]. However, in our experience this does not improve performance unless weights for the losses belonging to different machine types are manually tuned to improve the ASD performance. Since this requires access to anomalous samples, tuning these weights is impossible in a truly semi-supervised setting.

It can be seen that for both datasets the ASD performance improves with the number of classes being used for the auxiliary task. When using only a single class for all data or for individual machine types and sections, the AUC is close to 50%, which corresponds to randomly guessing whether a sample is anomalous or not. The most likely reason for this is the factory background noise contained in the recordings, which is highly diverse and contains many sound sources other than the target machine. A model trained with a one-class loss does not know the difference between the sound events emitted by the machines to be monitored and any other sounds contained in the recordings. The more complex (in terms of numbers of classes) the chosen auxiliary task is, the more information needs to be captured inside the embeddings for solving this task. Additionally, the background noise does not contain any helpful information for learning to discriminate between the classes defined by the auxiliary task assuming the noise is not class-specific. As a result, the model learns to monitor specific frequencies or temporal patterns important for specific machine types with specific settings and thus also learns to ignore the background noise and to isolate sounds emitted by the targeted machines. Furthermore, it can be observed that using an explicit classification task improves performance on all dataset splits. Ensuring an angular margin between the classes slightly improves the overall performance, but not significantly, often leading to very similar results. The most likely reason is that by increasing intra-class similarity implicitly introduces a margin between different classes. Still, using an angular margin loss does not have any drawbacks over using a compactness and a descriptiveness loss. As a last observation, the sub-cluster AdaCos loss performs slightly better than the AdaCos loss on the development split of the DCASE2022 dataset while yielding a similar performance on the other dataset splits. A possible explanation that there are no significant improvements on the DCASE2023 datasets when using an angular margin loss is that the auxiliary classification task is not as difficult as for the DCASE2022 dataset because

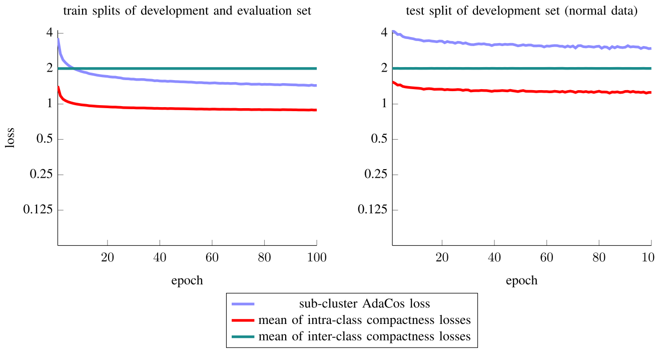


Fig. 2. Different losses after each epoch when training by minimizing sub-cluster AdaCos with a single sub-cluster per class and using mixup.

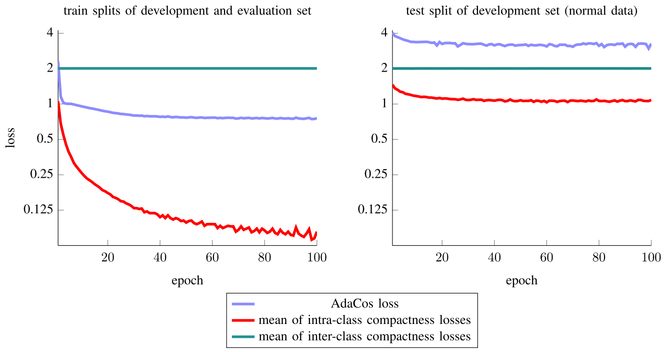


Fig. 3. Different losses after each epoch when training by minimizing AdaCos and not using mixup.

there is only one section for each machine type. Slight improvements in performance when using multiple sub-clusters for the AdaCos loss have been observed on the DCASE2020 dataset [2] in [29]. Note that the DCASE2020 dataset only contains machine recordings with a single parameter setting for each section and no domain shifts, i.e. consists of a single source domain, and thus the task is very different from the much more difficult task considered here. In conclusion, an angular margin loss for ASD in combination with an auxiliary classification task that uses as many meaningful classes as possible is an excellent choice when training an ASD system based on audio embeddings.

In the previous paragraph, we made the assumption that the noise is not class-specific. However, if there is a single class with very specific noise that is only present for this particular class or, even worse, if this is the case for all classes, then an auxiliary classification task will very likely not improve the results. The reason is that the model does not learn to closely monitor the machine sound because also the background noise contains useful information for discriminating between the classes. Therefore, assuming that the noise is not class-specific is essential and intuitively makes sense for machine condition monitoring as one would expect that at least some machines share the same noise distribution when running in the same factory or acoustic environment. Moreover, as shown in Theorem 6, minimizing an angular margin loss using an auxiliary classification task also explicitly increases intra-class similarity. Hence, even if the noise is class-specific and thus the auxiliary classification task does not aid the ASD task, the performance is still as least as good as when not using a classification task at all but only minimizing the intra-class compactness losses and there should not be a disadvantage.

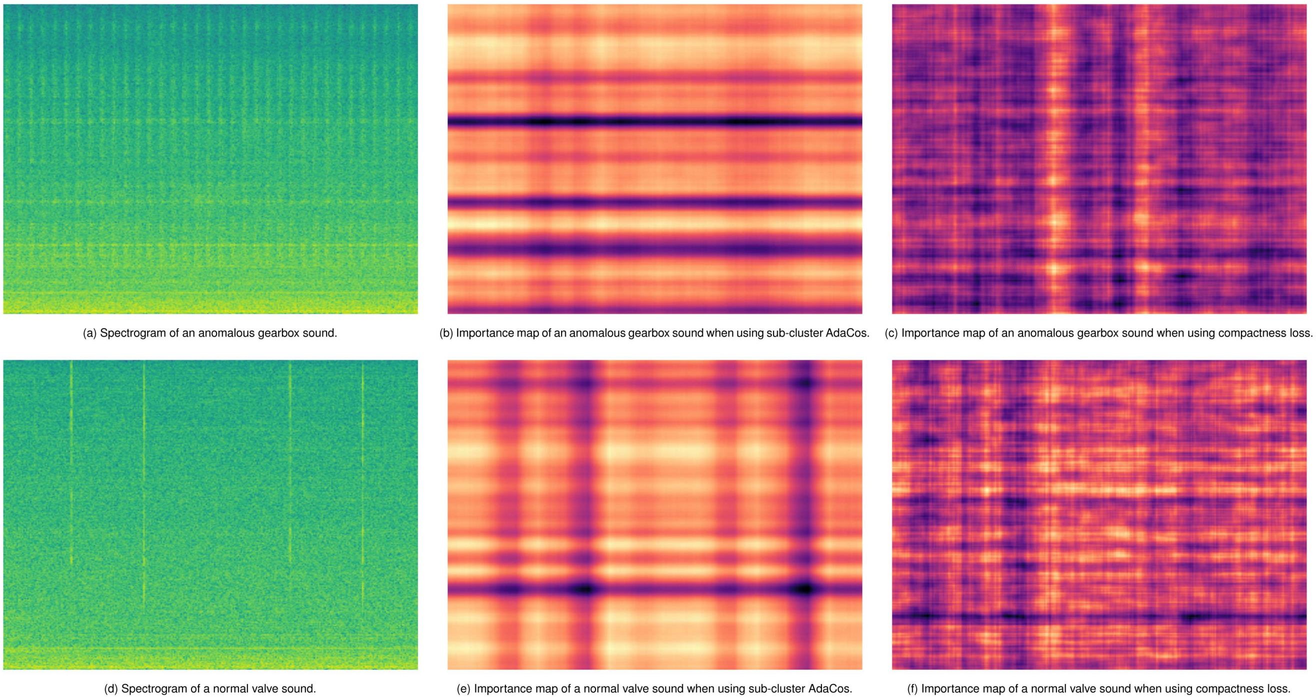
D. Minimizing Compactness Loss by Minimizing an Angular Margin Loss

In Theorem 6, it has been shown that minimizing an angular margin loss also minimizes all IC compactness losses and maximizes all inter-class compactness losses. This fact is now verified experimentally by training a model using the sub-cluster AdaCos loss while also monitoring all compactness losses. The results are depicted in Figs. 2 and 3. Regardless of the dataset splits and regardless of using or not using mixup, the angular margin loss and the mean of the IC compactness losses are decreasing during training. The mean of the inter-class compactness loss is constantly equal to 2, even without training. The reason is that all sub-cluster centers in this work are constant, randomly initialized and projected to the unit sphere. Hence, By Lemma 5, a squared Euclidean distance of 2 corresponds to an angle of $\frac{\pi}{2}$, i.e. orthogonality. The most likely reason is that the randomly initialized center vectors are approximately orthogonal with very high probability because of the high dimension $D = 256$ of the embedding space. Thus, samples that are similar to the center of one class will be approximately orthogonal to the centers of the other classes. Overall, this is exactly the expected behavior as predicted by Theorem 6 and therefore verifies the theoretical results. Note that smaller loss values do not correspond to a better ASD performance because minimizing these losses only optimizes the performance for the auxiliary task, which is not the same as the ASD task.

E. Visualizing Normal and Anomalous Regions in Input Representations as Perceived by the System

To further investigate the effect of using an auxiliary task with multiple classes, another experiment using RISE [74] is carried out. RISE highlights regions of the input representations that are considered normal or anomalous by the ASD system. Our goal is to show that utilizing an auxiliary classification task for training the system, as done when minimizing an angular margin loss, enables the system to closely monitor specific machine sounds by focusing on regions belonging to specific patterns of the input data. Although the ASD performance is worse when only using spectrograms as input representations [33], for these experiments a model using only spectrograms as input has been trained. The reason is that these representations are visually more appealing for the human eye than waveforms or spectra and thus more suitable to visually highlight normal and anomalous regions.

To visualize areas of the input representation responsible for a decision, RISE masks random entries of the spectrograms using binary masks and evaluates the ASD score using the masked spectrogram. This step is repeated for many iterations. Then, the sum of the masks weighted with the corresponding ASD scores is taken and normalized with the expected value of a random binary mask, which depends on the chosen sampling distribution. The result is called an *importance map* and visualizes the impact



(a) Spectrogram of an anomalous gearbox sound.

(b) Importance map of an anomalous gearbox sound when using sub-cluster AdaCos.

(c) Importance map of an anomalous gearbox sound when using compactness loss.

(d) Spectrogram of a normal valve sound.

(e) Importance map of a normal valve sound when using sub-cluster AdaCos.

(f) Importance map of a normal valve sound when using compactness loss.

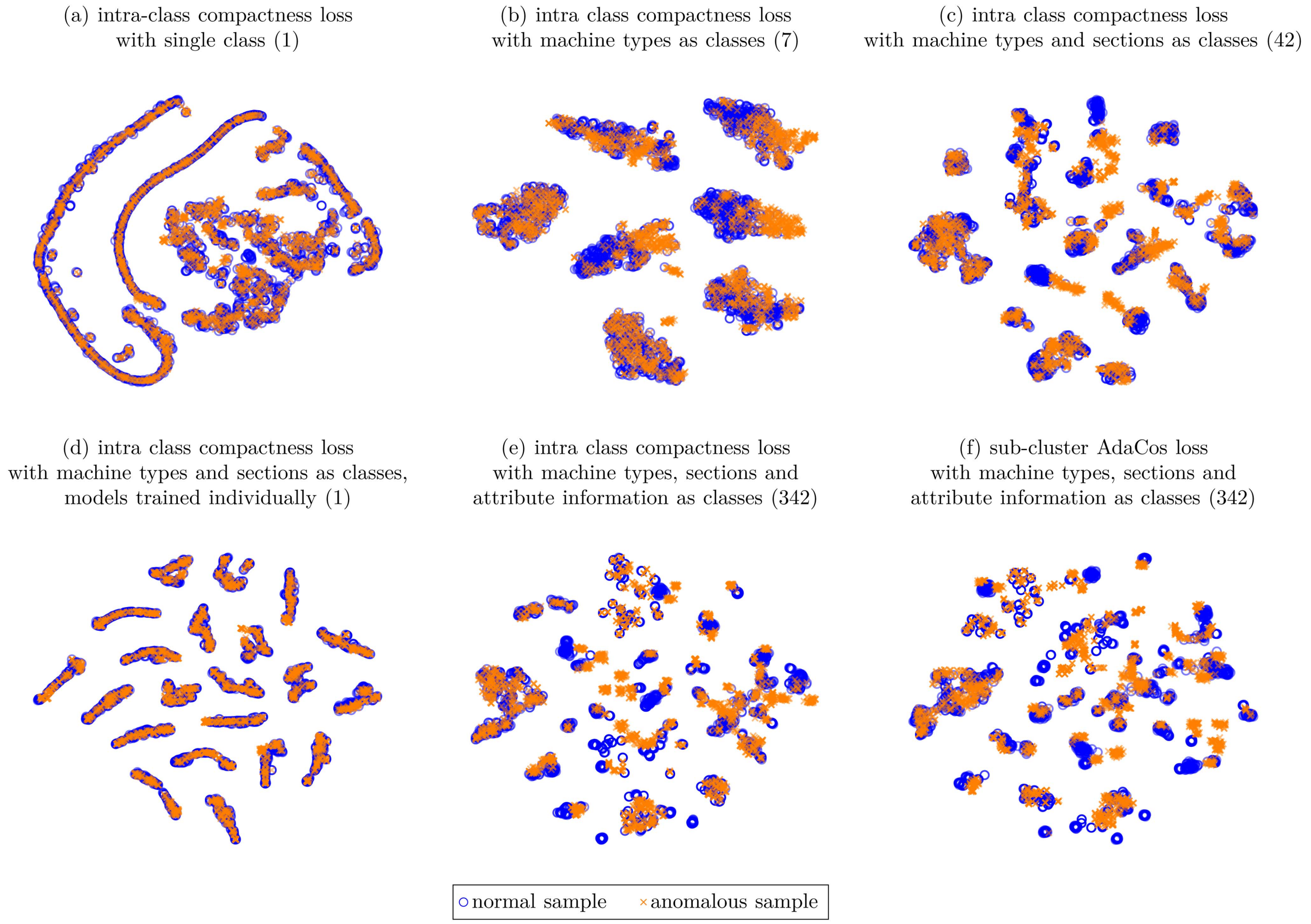
Fig. 4. Log scaled spectrograms (left column), importance maps obtained with RISE when training with the sub-cluster AdaCos loss and classifying between different machine types, sections and attribute information (middle column), and importance maps obtained with RISE when training with an IC compactness loss and no auxiliary classification task (right column) for two different recordings belonging to the test split of the development set (rows). For the importance maps, blue colors indicate normal regions and yellow colors indicate regions that are found to be anomalous by the model. All subfigures use individual color scales to improve visual appearance for differently scaled importance maps and thus colors of different subfigures cannot be compared to each other. (a) Spectrogram of an anomalous gearbox sound (b) Importance map of an anomalous gearbox sound when using sub-cluster AdaCos (c) Importance map of an anomalous gearbox sound when using compactness loss. (d) Spectrogram of a normal valve sound (e) Importance map of a normal valve sound when using sub-cluster AdaCos (f) Importance map of a normal valve sound when using compactness loss.

of specific regions of a spectrogram on the resulting anomaly score.

The problem is that the dimension of the spectrograms is very high because a time dimension of $T = 311$ and a frequency dimension of $F = 513$ is used. Thus, there are $2^{T\cdot F} = 2^{159543}$ possible binary masks and thus RISE requires clearly too many iterations. To significantly reduce the search space from $2^{F\cdot T}$ to 2^{F+T}, individual time and frequency masks are randomly generated with a probability of 0.25 for a time step or frequency bin to be masked and both masks are combined by element-wise multiplication. This restriction is not too severe because most sounds emitted by machines are relatively stable over time with specific frequencies (e.g. fans), consist of multiple stable sound events with on- and offsets (e.g. slide rails) or only consist of short sound events over a wide frequency range with a specific temporal structure (e.g. valves). For further reduction of the search space, small binary masks are generated and then up-sampled and randomly cropped to match the dimension of the spectrogram to be masked as proposed in [74]. More concretely, we used time masks of size 20 and frequency masks of size 34 resulting in a search space of 2^{54}, which is still very large but much smaller than before. For generating a single importance map, 640,000 iterations have been used.

Magnitude spectrograms (visualized in log scale) and corresponding importance maps belonging to two different samples using i) a model trained with an IC compactness loss without an auxiliary task, and ii) a model trained with the sub-cluster AdaCos loss and an auxiliary task for classifying between different machine types, sections and attribute information are depicted in Fig. 4. For the depicted importance maps, blue colors indicate normal regions and yellow colors indicate anomalous regions as perceived by the system. Note that, since the system does not yield perfect results, these regions do not need to really belong to normal and anomalous regions. As there are only binary labels, indicating normal or anomalous samples, available for each entire audio recording and we are no subject matter experts for machine condition monitoring, we do not know which regions are normal or anomalous. Still, for the purpose of showing that utilizing meta information when training a model, as done by angular margin losses, helps the system to have a better understanding of the structure of the data these plots are sufficient. There are several observations to be made. Comparing the representations depicted in Fig. 4(b) and (e) with the ones depicted in Fig. 4(c) and (f), we suggest that using sub-cluster AdaCos, i.e. Fig. 4(b) and (e), more clearly shows time and frequency structures at a resolution correlating with the structures resp. acoustic events visible in the spectrograms depicted in Fig. 4(a) and (d).

For the anomalous gearbox example (Fig. 4(a)), the importance map depicted in Fig. 4(b) shows that specific frequencies are monitored and considered to be normal or anomalous. Interestingly, the normal frequency regions (in blue) in Fig. 4(b)


(a) intra-class compactness loss with single class (1)

(b) intra class compactness loss with machine types as classes (7)

(c) intra class compactness loss with machine types and sections as classes (42)

(d) intra class compactness loss with machine types and sections as classes, models trained individually (1)

(e) intra class compactness loss with machine types, sections and attribute information as classes (342)

(f) sub-cluster AdaCos loss with machine types, sections and attribute information as classes (342)

Fig. 5. Visualizations of the test split of the development set in the learned embedding space for different loss functions and auxiliary tasks using t-distributed stochastic neighbor embedding (t-SNE). Numbers in brackets denote the number of different classes used for the auxiliary task.

exactly correspond to the frequencies containing high energy (Fig. 4(a)) showing that the model expects a gearbox sound from this section to have high energy in these regions. The frequencies that are considered most anomalous, which mostly corresponds to the frequency range between the bottom two normal frequency bands, only contain some energy. This indicates that a normal machine sound should either contain no energy or much more energy for these frequencies. In contrast to this, the importance map depicted in Fig. 4(c) does not monitor specific frequencies and the only clearly visible structures are two vertical lines indicating anomalous regions (in yellow). Although we cannot guarantee that the regions in the spectrogram corresponding to these vertical lines are not anomalous, at least visually there is no energy present in these locations. Since the recordings of the machine sounds do not start and end at the same fixed time steps, it does not make sense that the model expects temporal patterns at exactly these time steps that are missing and to thus consider such patterns to be anomalous. Therefore, it seems that these structures are errors of the model.

The importance maps belonging to the normal valve example (Fig. 4(d)) show a similar behavior but for temporal patterns in addition to specific frequencies. Here, the main four normal vertical patterns in the importance map shown in Fig. 4(e) correspond the four high energy patterns of the spectrogram showing that the system views these temporal patterns as normal for a valve sound. In contrast, the importance map depicted in Fig. 4(f) does not show that the system has learned to detect these patterns and looks almost random.

Overall, the depicted results add further confidence to the claim that training a model with an auxiliary classification task with many classes enables the model to learn much more meaningful embeddings, also leading to much better capabilities for detecting anomalous sound events than a model trained with only a single class.

F. Visualizing the Resulting Embedding Spaces Using t-SNE

As a last experiment, the embedding spaces resulting from using different loss functions and auxiliary tasks are visualized in Fig. 5 using t-SNE [75]. Note that by Lemma 5 it does not matter whether t-SNE is evaluated with the cosine distance or the Euclidean distance because both are equivalent when determining the degree of similarity between samples on the unit sphere. It can be seen that using more classes for the auxiliary task helps to separate normal and anomalous samples (Fig. 5(b), (c), (e), and (f)). When only using a single class (Fig. 5(a)) or individually

TABLE III
MEAN AND STANDARD DEVIATION OF THE AVERAGE EUCLIDEAN DISTANCE BETWEEN THE T-SNE PROJECTIONS OF EACH ANOMALOUS SAMPLE AND THE CLOSEST NORMAL SAMPLE OVER FIVE TRIALS FOR DIFFERENT LOSSES AND USING DIFFERENT AUXILIARY TASKS

loss	classes of auxiliary task (number of classes)	average distance
IC compactness loss	none (1)	0.485 ± 0.007
IC compactness loss	machine types (7)	1.636 ± 0.037
IC compactness loss	machine types and sections (42)	2.175 ± 0.075
IC compactness loss	machine types and sections, models trained individually (1)	0.559 ± 0.002
IC compactness loss	machine types, sections and attribute information (342)	2.646 ± 0.045
sub-cluster AdaCos loss	machine types, sections and attribute information (342)	2.947 ± 0.022

trained models (Fig. 5(d)), there is no visual difference between normal and anomalous samples. However, it can also be seen that the model has not learned a trivial solution as the embedding spaces did not collapse to a single fixed point, which would correspond to a uniformly distributed t-SNE embedding space. Moreover, the ASD performance would be very close to 50% as normal and anomalous samples would be indistinguishable in the embedding space. Therefore, the applied regularization strategies, namely not using trainable centers and not using bias terms, work and a completely failed regularization is not the main underlying problem. These visual impressions are verified by computing the average Euclidean distance between each anomalous sample and the closest normal sample in the t-SNE embedding space. The results can be found in Table III and also agree with the performance results shown in Table II. Note that the distance in the original embedding space is implicitly captured by the ASD performance given in Table II because the anomaly score is computed by taking the distance to the closest normal sample in the target domain and the closest mean in the source domain. Again, the most likely explanation for the strong differences between the embedding spaces in terms of ASD capabilities is that using multiple classes enables the model to focus less on or even ignore the background noise and isolate the targeted machine sounds. This helps the model to more robustly detect deviations from normal machine sounds despite the acoustically noisy recording conditions and thus results in better ASD performance.

V. Conclusion

In this work, it has been investigated why using angular margin losses works well for semi-supervised ASD. To this end, it has been shown, both theoretically and experimentally, that reducing an angular margin loss also minimizes the IC compactness loss while simultaneously maximizing the inter-class compactness loss. Therefore, angular margin losses in combination with an auxiliary classification task can be viewed as regularized one-class losses preventing the model to learn trivial solutions. In experiments conducted on the DCASE2022 and DCASE2023 ASD datasets for machine condition monitoring, it has been shown that using an auxiliary task with as many meaningful classes as possible and using an angular margin loss leads to significantly better ASD performance than using a one-class loss such as the IC compactness loss. Furthermore, RISE has been applied to create importance maps for different losses and t-SNE has been used to visualize the resulting embedding spaces. All the conducted experiments show that by using an angular margin the model used for extracting the embeddings learns to monitor relevant frequency bins and learns machine-specific temporal patterns. This enables the model to isolate machine sounds and effectively ignore background noise present in the recording explaining why angular margin losses with an auxiliary task are a good choice for training an ASD system.

For future work, is is planned to investigate whether using auxiliary tasks based on self-supervised learning to obtain suitable representations of the data improves the resulting ASD performance. In addition, sophisticated methods for visualizing anomalous regions of input representations should be developed as being able to localize these regions is very useful for practical applications and theoretical analysis of ASD systems.

Appendix

Proof of Lemma 5

Using only basic definitions, we obtain

$$\begin{aligned}\|x-y\|_2^2 &= \sum_{i=1}^{D}(x_i-y_i)^2\\ &= \sum_{i=1}^{D}x_i^2+\sum_{i=1}^{D}y_i^2-2\sum_{i=1}^{D}x_iy_i\\ &= \|x\|_2^2+\|y\|_2^2-2\langle x,y\rangle\\ &= 2\left(1-\frac{\langle x,y\rangle}{\|x\|_2\|y\|_2}\right)\\ &= 2(1-\cos(x,y)),\end{aligned}$$

which finishes the proof. □

Acknowledgment

We would like to thank Paul M. Baggenstoss and Lukas Henneke as well as the anonymous reviewers for their valuable comments that improved the quality of this work.

References

[1] C. Aggarwal, *Outlier Analysis*, 2nd ed. Berlin, Germany: Springer, 2017.

[2] Y. Koizumi et al., “Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring,” in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2020, pp. 81–85.

[3] Y. Kawaguchi et al., “Description and discussion on DCASE 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions,” in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2021, pp. 186–190.

[4] K. Dohi et al., “Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques,” in *Proc. 7th Workshop Detection Classification Acoust. Scenes Events*, 2022, pp. 26–30.

[5] S. N. Murthy and E. Agu, “Deep learning anomaly detection methods to passively detect COVID-19 from audio,” in *Proc. IEEE Int. Conf. Digit.*, 2021, pp. 114–121.

[6] T. Dissanayake, T. Fernando, S. Denman, S. Sridharan, H. Ghaemmaghami, and C. Fookes, “A robust interpretable deep learning classifier for heart anomaly detection without segmentation,” *IEEE J. Biomed. Health Inform.*, vol. 25, no. 6, pp. 2162–2171, Jun. 2021.

[7] S. Ntalampiras and I. Potamitis, “Acoustic detection of unknown bird species and individuals,” *CAAI Trans. Intell. Technol.*, vol. 6, no. 3, pp. 291–300, 2021.

[8] T. Cejrowski and J. Szymanski, "Buzz-based honeybee colony fingerprint," *Comput. Electron. Agriculture*, vol. 191, 2021, Art. no. 106489.

[9] C. Zieger, A. Brutti, and P. Svaizer, "Acoustic based surveillance system for intrusion detection," in *Proc. IEEE 6th Int. Conf. Adv. Video Signal Based Surveill.*, 2009, pp. 314–319.

[10] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance of roads: A system for detecting anomalous sounds," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 279–288, Jan. 2016.

[11] Y. Li, X. Li, Y. Zhang, M. Liu, and W. Wang, "Anomalous sound detection using deep audio representation and a BLSTM network for audio surveillance of roads," *IEEE Access*, vol. 6, pp. 58043–58055, 2018.

[12] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *Proc. 4th Int. Conf. Adv. Video Signal Based Surveill.*, 2007, pp. 21–26.

[13] T. Hayashi, T. Komatsu, R. Kondo, T. Toda, and K. Takeda, "Anomalous sound event detection based on wavenet," in *Proc. IEEE 26th Eur. Signal Process. Conf.*, 2018, pp. 2494–2498.

[14] S. Shon, N. Dehak, D. A. Reynolds, and J. R. Glass, "MCE 2018: The 1st multi-target speaker detection and identification challenge evaluation," in *Proc. 20th Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 356–360.

[15] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," in *Proc. Workshop Detection Classification Acoust. Scenes Events*, 2019, pp. 164–168.

[16] J. Naranjo-Alcazar et al., "An open-set recognition and few-shot learning dataset for audio event classification in domestic environments," *Pattern Recognit. Lett.*, vol. 164, pp. 40–45, 2022.

[17] E. Marchi, F. Vesperini, S. Squartini, and B. W. Schuller, "Deep recurrent neural network-based autoencoders for acoustic novelty detection," *Comput. Intell. Neurosci.*, vol. 2017, 2017, Art. no. 4694860.

[18] 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, Jan. 2019.

[19] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 271–275.

[20] 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. 5th Workshop Detection Classification Acoust. Scenes Events*, 2020, pp. 51–55.

[21] S. Kapka, "ID-conditioned auto-encoder for unsupervised anomaly detection," in *Proc. 5th Workshop Detection Classification Acoust. Scenes Events*, 2020, pp. 71–75.

[22] G. Wichern, A. Chakrabarty, Z. Wang, and J. Le Roux, "Anomalous sound detection using attentive neural processes," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2021, pp. 186–190.

[23] K. Dohi, T. Endo, H. Purohit, R. Tanabe, and Y. Kawaguchi, "Flow-based self-supervised density estimation for anomalous sound detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 336–340.

[24] K. Dohi, T. Endo, and Y. Kawaguchi, "Disentangling physical parameters for anomalous sound detection under domain shifts," in *Proc. IEEE 30th Eur. Signal Process. Conf.*, 2022, pp. 279–283.

[25] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Proc. 5th Workshop Detection Classification Acoust. Scenes Events*, 2020, pp. 46–50.

[26] J. A. Lopez, H. Lu, P. Lopez-Meyer, L. Nachman, G. Stemmer, and J. Huang, "A speaker recognition approach to anomaly detection," in *Proc. 5th Workshop Detection Classification Acoust. Scenes Events*, 2020, pp. 96–99.

[27] T. Inoue et al., "Detection of anomalous sounds for machine condition monitoring using classification confidence," in *Proc. 5th Workshop Detection Classification Acoust. Scenes Events*, 2020, pp. 66–70.

[28] Q. Zhou, "ArcFace based sound mobilenets for DCASE 2020 task 2," DCASE2020 Challenge, Tech. Rep., 2020.

[29] K. Wilkinghoff, "Sub-cluster AdaCos: Learning representations for anomalous sound detection," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2021, pp. 1–8.

[30] K. Wilkinghoff, "Combining multiple distributions based on sub-cluster AdaCos for anomalous sound detection under domain shifted conditions," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2021, pp. 55–59.

[31] S. Venkatesh, G. Wichern, A. S. Subramanian, and J. Le Roux, "Improved domain generalization via disentangled multi-task learning in unsupervised anomalous sound detection," in *Proc. 7th Workshop Detection Classification Acoust. Scenes Events*, 2022, pp. 196–200.

[32] T. Nishida, K. Dohi, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Anomalous sound detection based on machine activity detection," in *Proc. IEEE 30th Eur. Signal Process. Conf.*, 2022, pp. 269–273.

[33] K. Wilkinghoff, "Design choices for learning embeddings from auxiliary tasks for domain generalization in anomalous sound detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[34] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," in *Proc. 7th Int. Conf. Learn. Representations*, 2019, pp. 1–18.

[35] 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. 5th Workshop Detection Classification Acoust. Scenes Events*, 2020, pp. 170–174.

[36] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6738–6746.

[37] H. Wang et al., "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5265–5274.

[38] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4690–4699.

[39] S. Grollmisch, D. Johnson, J. Abeßer, and H. Lukashevich, "IAEO3-combining OpenL3 embeddings and interpolation autoencoder for anomalous sound detection," DCASE2020 Challenge, Tech. Rep., 2020 .

[40] K. Wilkinghoff, "Using look, listen, and learn embeddings for detecting anomalous sounds in machine condition monitoring," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, 2020, pp. 215–219.

[41] R. Müller, F. Ritz, S. Illium, and C. Linnhoff-Popien, "Acoustic anomaly detection for machine sounds based on image transfer learning," in *Proc. 13th Int. Conf. Agents Artif. Intell.*, 2021, pp. 49–56.

[42] K. Wilkinghoff and F. Fritz, "On using pre-trained embeddings for detecting anomalous sounds with limited training data," in *Proc. IEEE 31st Eur. Signal Process. Conf.*, 2023, pp. 186–190.

[43] J. A. Lopez, G. Stemmer, P. Lopez-Meyer, P. Singh, J. A. del Hoyo Ontiveros, and H. A. Cordourier, "Ensemble of complementary anomaly detectors under domain shifted conditions," in *Proc. 6th Workshop Detection Classification Acoust. Scenes Events*, 2021, pp. 11–15.

[44] I. Kuroyanagi, T. Hayashi, Y. Adachi, T. Yoshimura, K. Takeda, and T. Toda, "An ensemble approach to anomalous sound detection based on conformer-based autoencoder and binary classifier incorporated with metric learning," in *Proc. 6th Workshop Detection Classification Acoust. Scenes Events*, 2021, pp. 110–114.

[45] Y. Deng et al., "Ensemble of multiple anomalous sound detectors," in *Proc. 7th Workshop Detection Classification Acoust. Scenes Events*, 2022, pp. 21–25.

[46] A. Fernandez and M. D. Plumbley, "Using UMAP to inspect audio data for unsupervised anomaly detection under domain-shift conditions," in *Proc. 6th Workshop Detection Classification Acoust. Scenes Events*, 2021, pp. 165–169.

[47] A. Holzinger, A. Saranti, C. Molnar, P. Biecek, and W. Samek, "Explainable AI methods – A brief overview," in *Proc. Int. Workshop Extending Explainable AI Beyond Deep Models Classifiers*, 2020, pp. 13–38.

[48] K. T. Mai, T. Davies, L. D. Griffin, and E. Benetos, "Explaining the decision of anomalous sound detectors," in *Proc. 7th Workshop Detection Classification Acoust. Scenes Events*, 2022, pp. 1–5.

[49] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?" Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.

[50] S. Mishra, B. L. Sturm, and S. Dixon, "Local interpretable model-agnostic explanations for music content analysis," in *Proc. 18th Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 537–543.

[51] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv: 1802.03426.*

[52] L. Ruff et al., "Deep one-class classification," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 4390–4399.

[53] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[54] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[55] P. Perera and V. M. Patel, "Learning deep features for one-class classification," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5450–5463, Nov. 2019.

[56] A. N. Gorban, I. Y. Tyukin, D. V. Prokhorov, and K. I. Sofeikov, "Approximation with random bases: Pro et contra," *Inf. Sci.*, vol. 364–365, pp. 129–145, 2016.

[57] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[58] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. 6th Int. Conf. Learn. Representations*, 2018, pp. 1–13.

[59] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "AdaCos: Adaptively scaling cosine logits for effectively learning deep face representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10823–10832.

[60] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.

[61] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.

[62] J. S. Chung et al., "In defence of metric learning for speaker recognition," in *Proc. 21st Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2977–2981.

[63] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, 2021, Art. no. 63.

[64] K. Wilkinghoff and A. Cornaggia-Urrigshardt, "TACos: Learning temporally structured embeddings for few-shot keyword spotting with dynamic time warping," 2023, *arXiv:2305.10816*.

[65] K. Dohi et al., "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," in *Proc. 7th Detection Classiciation Acoust. Scenes Events Workshop*, 2022, pp. 31–35.

[66] 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," in *Proc. 6th Detection Classification Acoust. Scenes Events Workshop*, 2021, pp. 1–5.

[67] J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 2021, pp. 4627–4635.

[68] K. Dohi et al., "Description and discussion on DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. 8th Detection Classification Acoust. Scenes Events Workshop*, 2023, pp. 31–35.

[69] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "First-shot anomaly detection for machine condition monitoring: A domain generalization baseline," in *Proc. 31st Eur. Signal Process. Conf.*, 2023, pp. 191–195.

[70] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "ToyADMOS2+: New Toyadmos data and benchmark results of the first-shot anomalous sound event detection baseline," in *Proc. 8th Detection Classification Acoust. Scenes Events Workshop*, 2023, pp. 41–45.

[71] D. K. McClish, "Analyzing a portion of the ROC curve," *Med. Decis. Mak.*, vol. 9, no. 3, pp. 190–195, 1989.

[72] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 1021–1025.

[73] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Des. Implementation*, 2016, pp. 265–283.

[74] V. Petsiuk, A. Das, and K. Saenko, "RISE: Randomized input sampling for explanation of black-box models," in *Proc. Brit. Mach. Vis. Conf.*, 2018, Art. no. 151.

[75] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2431–2456, 2008.

Kevin Wilkinghoff (Student Member, IEEE) received the B.Sc. degree in mathematics from the University of Münster, Münster, Germany, in 2014, and the M.Sc. degree in computer science in 2017 from the University of Bonn, Bonn, Germany, where he is currently working toward the Ph.D. degree in computer science. Since 2017, he has been a Research Associate with Fraunhofer FKIE, Wachtberg, Germany. His research interests include anomaly detection, open-set classification, and representation learning for machine listening applications. In 2021, he was the recipient of the DCASE Best Paper Award.

Frank Kurth (Senior Member, IEEE) studied Computer Science and Mathematics and received the master's degree in computer science, the degree of a doctor of natural sciences (Dr. rer. nat.), and the Habilitation degree in computer science from Bonn University, Bonn, Germany, in 1997, 1999, and 2004, respectively. From 1997 to 2007, he was with the Multimedia Signal Processing Group, Bonn University where he was an appointed apl. Professor in 2013. Since 2007, he has been with Fraunhofer FKIE, Wachtberg, Germany, where he currently heads a research group focused on physical layer signal analysis in the area of communications. He is the co-author of more than 100 publications and holds several patents. His research interests include the application of pattern recognition and machine learning techniques to audio, speech and communication signal processing. Dr. Kurth was the recipient of the 2000 Dissertation Award of the German Informatics Society (GI) and 2000 Multimedia Award of the German Department of Economy and Technology.