Hyperbolic Unsupervised Anomalous Sound Detection

Germain, Francois; Wichern, Gordon; Le Roux, Jonathan

TR2023-108 September 02, 2023

Abstract

We introduce a framework to perform unsupervised anomalous sound detection by leveraging embeddings learned in hyperbolic space. Previously, hyperbolic spaces have demonstrated the abil- ity to encode hierarchical relationships much more effectively than Euclidean space when using those embeddings for classification. A corollary of that property is that the distance of a given embedding from the hyperbolic space origin encodes a notion of classification certainty, naturally mapping inlier class samples to the space edges and outliers near the origin. As such, we expect the hyperbolic em- beddings generated by a deep neural network pre-trained to classify short-time Fourier transform frames of normal machine sounds to be more distinctive than Euclidean embeddings when attempting to identify unseen anomalous data. In particular, we show here how to perform unsupervised anomaly detection using embeddings from a trained modified MobileFaceNet architecture with a hyperbolic em- bedding layer, using the embeddings generated from a test sample to generate an anomaly score. Our results show that the proposed approach outperforms similar methods in Euclidean space on the DCASE 2022 Unsupervised Anomalous Sound Detection dataset.

IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) 2023

HYPERBOLIC UNSUPERVISED ANOMALOUS SOUND DETECTION

François G. Germain, Gordon Wichern, Jonathan Le Roux

Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

ABSTRACT

We introduce a framework to perform unsupervised anomalous sound detection by leveraging embeddings learned in hyperbolic space. Previously, hyperbolic spaces have demonstrated the ability to encode hierarchical relationships much more effectively than Euclidean space when using those embeddings for classification. A corollary of that property is that the distance of a given embedding from the hyperbolic space origin encodes a notion of classification certainty, naturally mapping inlier class samples to the space edges and outliers near the origin. As such, we expect the hyperbolic embeddings generated by a deep neural network pre-trained to classify short-time Fourier transform frames of normal machine sounds to be more distinctive than Euclidean embeddings when attempting to identify unseen anomalous data. In particular, we show here how to perform unsupervised anomaly detection using embeddings from a trained modified MobileFaceNet architecture with a hyperbolic embedding layer, using the embeddings generated from a test sample to generate an anomaly score. Our results show that the proposed approach outperforms similar methods in Euclidean space on the DCASE 2022 Unsupervised Anomalous Sound Detection dataset.

Index Terms— anomalous sound detection, hyperbolic space, machine sound, surrogate task

1. INTRODUCTION

Automatically detecting faulty equipment, i.e., anomaly detection [1], is an essential task in the modern industrial society. Performing such detection from sound, i.e., anomalous sound detection, is especially appealing due to factors such as sensor cost and ability to measure signals without line of sight. Audio or not, practical anomaly detection design is hampered by the difficulty of collecting anomalous samples, which, beyond the cost of labeling, is further affected by issues such as the rare occurrence of anomalies or the cost associated with deliberately provoking them. As such, unsupervised approaches are of particular interest in the field. Anomalous sound detection branched out of general sound event detection before growing into its own field [2–4]. Since 2020, unsupervised detection has even become a staple of the yearly Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge [5–8].

One popular category of approaches is surrogate-task (ST) methods [9], which have been fairly successful in the DCASE challenges, including as 2022 challenge baseline [7, 9–12]. It involves the basic approach of identifying a surrogate classification task for the normal data, followed by the training of a classifier on that data. We then consider the distribution of learned embeddings (i.e., the last hidden vector before the output logits in the classifier network) as a representation of normal data. An anomaly detector is then built on top of that learned distribution, using the relative position of an unseen sample's embedding with respect to the distribution to determine the likely condition, normal or anomalous. For example, the distance of the embedding to its K-nearest neighbors in the trained embedding distribution can be used as criterion, setting a distance threshold above which a sample is deemed anomalous [9].

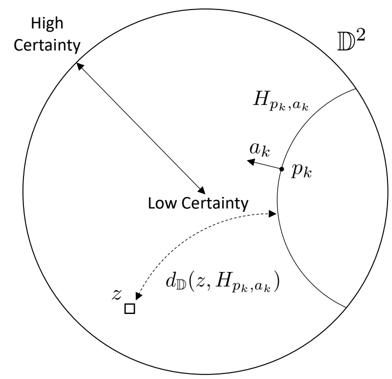

Figure 1: 2-D Poincaré ball. A hyperbolic hyperplane H_{p_k,a_k} is defined by a point p_k and a vector a_k [13]. For a given point z, the distance to H_{p_k,a_k} corresponds to geodesic distance $d_{\mathbb{D}}(z, H_{p_k,a_k})$. Embeddings near the origin have low certainty, and embeddings near the edges have high certainty [14].

We propose to train and analyse embeddings as vectors in hyperbolic space [13] rather than the typical vectors in Euclidean space. So-called hyperbolic neural networks have attracted interest in multiple fields, e.g., natural language [13, 15], image [14, 16], or graph modeling [17, 18]. The approach is practically appealing due to the fact that its geometric properties make it suitable to naturally encode the hierarchical aspects we expect to find in many audio tasks and datasets. Much recent research has attempted to surface and leverage such aspects [19–21], including through the use of hyperbolic neural networks [22–24]. A particularly appealing aspect in the context of ST anomaly detection methods is the corollary behavior of embeddings in hyperbolic space shown in [14] such that, as information gets organized hierarchically in space, the distance of an embedding to the origin expresses something akin to a notion of certainty regarding the characteristics of the input.

We then explore the benefits swapping in a hyperbolic space for learning embeddings in an ST-based method inspired by [9], showing it to be a simple and effective detection method.

2. HYPERBOLIC NEURAL NETWORKS

2.1. Hyperbolic spaces

Riemannian geometry generalizes Euclidean geometry, by which an n-D Riemannian manifold is any pair of an n-D differentiable manifold and a so-called metric tensor field. Following that theory, a Euclidean manifold is simply a differentiable manifold whose metric tensor field is the identity everywhere. On the other hand, a hyper-

Table 1: MobileFaceNet architecture adapted from [9, 26]. All convolutions are 2-D. *dw-Conv* refers to depth-wise convolution. For each layer, we show the expansion factor **t**, number of channels **c**, number of repeats **n**, and stride **s**. All convolutions excluding the final linear layers use PReLU as the non-linearity.

Input	Operator	t	c	n	s
1×32×1025	Conv 3×3	-	64	1	2
64×16×513	dw-Conv 3×3	-	64	1	1
64×16×513	Bottleneck	2	64	5	2
64×8×257	Bottleneck	4	128	1	2
128×4×129	Bottleneck	2	128	6	2
128×2×65	Bottleneck	4	128	1	2
128×1×33	Bottleneck	2	128	2	1
128×1×33	Conv 1×1	-	512	1	1
512×1×33	Linear GDC 1×33	-	512	1	1
512×1×1	Linear Conv 1×1	-	L	1	1

bolic manifold is any Riemannian manifold with negative constant sectional curvature. Interestingly, even though hyperbolic spaces are not vector spaces in the traditional sense, recent literature has shown the ability to find equivalents in hyperbolic space to many typical vector operations found in deep learning [13, 14, 25].

Due to the impossibility of embedding isometrically a hyperbolic space into Euclidean space, we must in practice use models of hyperbolic geometry in which a subset of Euclidean space is endowed with a hyperbolic metric. One popular practical model is the n-D Poincaré ball with negative unit curvature, defined as the manifold inside the n-D unit ball $\mathbb{D}^n = \{\mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| < 1\}$ endowed with the metric tensor field $\frac{2}{1-\|\mathbf{x}\|^2}\mathbf{I}^n$ with $\mathbf{I}^n$ the identity. Fig. 1 shows the 2-D Poincaré ball. In this model, we know that the geodesic (i.e., shortest-path) distance between 2 points $\mathbf{x}$ and $\mathbf{y}$ is [13]

$$d_{\mathbb{D}}(\mathbf{x}, \mathbf{y}) = \cosh^{-1}\left(1 + 2\frac{\|\mathbf{x}-\mathbf{y}\|^2}{(1-\|\mathbf{x}\|^2)(1-\|\mathbf{y}\|^2)}\right). \quad (1)$$

One can then define a mapping, the "exponential map," from Euclidean space $\mathbb{R}^n$ to Poincaré ball space $\mathbb{D}^n$ by associating a vector $\mathbf{u} \in \mathbb{R}^n$ to vector $\mathbf{v} \in \mathbb{D}^n$ reached from $\mathbf{0} \in \mathbb{D}^n$ in unit time following the geodesic with initial tangent vector $\mathbf{u}$. The inverse mapping is the "logarithmic map." These mappings can be written [13]

$$\exp_{\mathbb{D}}(\mathbf{u}) = \tanh\|\mathbf{u}\| \cdot \frac{\mathbf{u}}{\|\mathbf{u}\|} \text{ and } \log_{\mathbb{D}}(\mathbf{v}) = \tanh^{-1}\|\mathbf{v}\| \cdot \frac{\mathbf{v}}{\|\mathbf{v}\|}. \quad (2)$$

2.2. Hyperbolic embeddings

A typical practice in deep learning is to designate the hidden vectors generated by some intermediate hidden layer, often the deepest, as embeddings. Embeddings obtained from trained classification neural networks have often been found to be useful representations of the input data, with the hope that their distribution will encode high-level characteristics of the data as the classification accuracy improves. Then, they have often been leveraged for downstream tasks different from the original classification task. In that spirit, various losses can be found in the literature to promote particular geometric characteristics in the distribution of those embeddings in Euclidean space (e.g., CosFace [27], SphereFace [28], ArcFace [29]).

Hyperbolic spaces possess geometric properties that make them specifically appealing in that context. For example, their volume grows exponentially as we get further from the origin, unlike in Euclidean space where volume grows polynomially. It is then possible to embed tree structures in a hyperbolic space with arbitrary

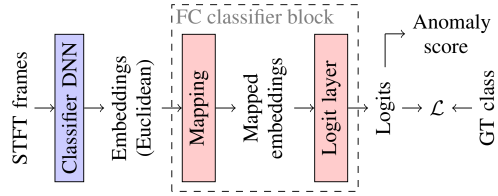

Figure 2: System diagram

low distortion [25]. Shallower tree nodes are then positioned closer to the origin, and deeper nodes farther. Equivalently, the geodesic distance between two points behaves similarly as the path length between two nodes in a tree. As such, hierarchical characteristics can be expected to be effectively encoded in that space. Concurrently, we generally expect high-level aspects of many typical datasets to exhibit natural hierarchies. Hence, prior research has found benefit in many applications in mapping the embeddings generated by a deep neural network to a hyperbolic space before performing the geometric equivalent of a multinomial regression in that space [25] using hyperplanes like the one in Fig. 1. Prior research [14] has also shown empirical evidence that using hyperbolic embeddings in a classifier results (after training) in a geometrical distribution of embeddings where the geodesic distance to the origin correlates well with a straightforward definition of certainty as a function of the predicted class probabilities. Note that since all vectors in the network except for these mapped embeddings are in Euclidean space, this type of approach is typically labelled as hybrid [15].

3. ANOMALY DETECTION

3.1. Data

We use the DCASE 2022 Task 2 challenge dataset [7]. We emphasize that our focus here is not on competing with the top ranked systems in the challenge, but rather on evaluating hyperbolic embeddings in a controlled setting. The data consists of normal and anomalous sounds recorded from 7 machine types with mixed environmental noise background. All recordings are single-channel, 10 s-long, and sampled at 16 kHz. For each machine type, we get 6 distinct "sections" corresponding to different machines of that type. Within each section, we are further given 2 distinct domains, "source" and "target," representing domain shifts such as differences in machine conditions. For training, we get the training data of the development subset and the additional training subset, i.e., normal-only data from all 6 sections, with 990 (resp. 10) normal samples from the source (resp. target) domain per section. For validation, we get the test data of the development subset, i.e., 50 samples of each 4 condition pairs in $\{\text{normal}, \text{anomalous}\} \times \{\text{source}, \text{target}\}$ per section for 3 of the sections. For evaluation, we are provided with the evaluation subset, i.e., 200 samples with the same condition proportions per section for the 3 remaining sections.

We process each file using an STFT with a 2048-sample Hann window and a 256-sample hop size, resulting in 313 frames from which we take the magnitude. For each epoch, we train the network with one block of 32 consecutive STFT frames from each file, i.e., 6000 blocks of size 32×1025, selected randomly for each file. We further group those blocks in batches of size 32. At testing, we break the magnitude STFT of a given file in overlapping blocks of 32 frames with a hop size of 1 frame. We then gather the embeddings and logits for all these blocks and compute the chosen scoring function to obtain the anomaly score for that file (see Sec. 3.4).

Table 2: Results on the DCASE 2022 Task 2 validation dataset. Each machine type columns shows the corresponding $\mathcal{S}_m$ for all 7 machine types. Hyperbolic systems are underlined. For each value of L, **bold** means best and *italic* means 2nd best system. w comes from Eq. (6).

L	System	Score	ToyCar	ToyTrain	Bearing	Fan	Gearbox	Slider	Valve	AUC (S)	AUC (T)	pAUC	Overall
2	Hyperbolic	$\mathcal{A}$	**60.1**	*57.0*	*61.9*	*71.2*	70.2	*81.6*	*80.2*	*68.9*	*71.1*	*63.6*	*67.7*
	Hyperbolic*	$\mathcal{A}_*$	48.2	55.1	57.7	66.3	61.4	79.3	83.4	62.7	63.8	60.7	62.4
	Ensemble	$\mathcal{A}_{\text{ens}}$	**60.1**	*57.0*	*61.9*	**71.4**	*70.4*	**83.0**	**83.5**	**69.0**	**71.9**	**64.3**	**68.2**
	(weights w)	—	*(0.0)*	*(0.0)*	*(0.0)*	*(0.4)*	*(0.4)*	*(0.1)*	*(0.9)*	—	—	—	—
	Euclidean	$\mathcal{A}$	59.2	50.6	**64.4**	65.8	**72.6**	77.4	75.7	66.0	69.2	61.1	65.2
	ArcFace	$\mathcal{A}$	58.9	**57.9**	61.4	60.6	62.9	78.0	78.7	*68.9*	63.0	62.1	64.5
128	Hyperbolic	$\mathcal{A}$	**60.8**	*54.6*	68.8	*67.3*	**75.2**	82.1	*83.4*	**72.2**	*69.7*	**65.1**	**68.9**
	Euclidean	$\mathcal{A}$	*58.6*	54.4	*73.0*	63.2	*71.8*	**83.3**	80.1	70.4	69.4	63.7	67.7
	ArcFace	$\mathcal{A}$	56.2	**56.3**	**73.5**	**69.8**	69.6	*82.6*	**84.2**	*70.9*	**71.1**	*64.6*	*68.7*

Table 3: Results on the DCASE 2022 Task 2 evaluation dataset. We are unable to give full results for the DCASE baseline in [7], as the necessary $\text{AUC}^{(T)}_{s,m}$, $\text{AUC}^{(S)}_{s,m}$, and $\text{pAUC}_{s,m}$ are not available.

L	System	AUC (S)	AUC (T)	pAUC	Overall
1280	DCASE ST [8]	59.1	47.5	53.6	53.0
—	DCASE AE [8]	64.5	45.2	52.9	53.1
2	Hyperbolic	**66.8**	58.8	58.0	*60.9*
	Hyperbolic*	61.3	56.0	*58.6*	58.6
	Ensemble	*66.3*	**61.5**	**58.7**	**62.0**
	Euclidean	62.9	*59.9*	57.6	60.0
	ArcFace	59.7	52.5	55.3	55.7
128	Hyperbolic	**66.0**	54.1	57.9	58.9
	Euclidean	*65.6*	**60.2**	**59.8**	**61.7**
	ArcFace	*65.6*	*58.1*	*59.3*	*60.8*

3.2. Models

Inspired by [9], the classifier backbone is an adapted version of the MobileFaceNet architecture[1] [26] described in Tab. 1. The main needed modification is in the global depthwise convolution (*Linear GDC*) layer where we use 1×33 kernels instead of 7×7 ones. MobileFaceNet is a lightweight version of MobileNetV2 [30], the architecture used in the ST-based DCASE 2022 baseline [7]. The salient feature of these networks are their bottleneck operators.

As seen in Fig. 2, the classifier backbone outputs embeddings readily interpretable as vectors in L-D Euclidean space. These are then passed to a fully-connected (FC) classifier block that outputs class logits. It is inside that block that we leverage hyperbolic space, mapping first the embeddings onto the Poincaré ball with $\exp_{\mathbb{D}}$ and then using a hyperbolic multinomial regression layer[2] [14].

As baselines, we train a system matching a regular multinomial regression classifier ("Euclidean"). It uses an identity mapping and its logit layer is a fully-connected layer with 6 output channels for the 6 sections per machine type. We also train a system using the popular ArcFace classifier block [29] ("ArcFace") for completeness. It also uses an identity mapping and its logit layer differs between training and testing in order to incentivize the emergence of preset margins in the (Euclidean) embedding space at training. For all systems, we train a version for $L = 2$ and $L = 128$.

3.3. Training

For both the proposed system and baselines, we train one classifier per machine type, each learning to recognize the section to which a given block of magnitude STFT frames belongs. We apply a cross-entropy loss to the output logits. Formally, for the ith input magnitude STFT block $\mathbf{X}^{(i)} \in \mathbb{R}_+^{32\times1025}$ generating an output logit vector $\mathbf{y}^{(i)} \in \mathbb{R}^6$ whose ground truth class/section is k_i, the loss is

$$\mathcal{L}(\mathbf{y}^{(i)}) = \log \frac{\exp y^{(i)}_{k_i}}{\sum_{k=1}^{6} \exp y^{(i)}_k}. \tag{3}$$

For our hyperbolic model, we use the Riemannian Adam optimizer[3] [31]. For the baselines, we use the PyTorch 1.10 Adam optimizer. The learning rate is 10^{-4}, other parameters are set to default. We train for 1000 epochs with checkpoints every 25 epochs.

3.4. Score and metrics

At test time, we use the trained network to output an anomaly score for an unseen audio file $\mathbf{x}$. We test the score from the ST-based baseline from the DCASE 2022 Task 2 challenge [7]. In that formulation, the score of a given file is based on the negative logit corresponding to the ground-truth section of that file. In the case where a given file is split in multiple segments, the score of the file becomes the segment-average of the score. In other words, if we denote $\psi_s(\mathbf{x}_k)$ the predicted probability that the kth segment of file $\mathbf{x}$, of ground-truth section t, belongs to section s, the score $\mathcal{A}$ is written as

$$\mathcal{A}(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} \log \left(\frac{1 - \psi_t(\mathbf{x}_k)}{\psi_t(\mathbf{x}_k)} \right). \tag{4}$$

For hyperbolic embeddings, we also experiment with using the negative segment-average geodesic distance to the origin in the Poincaré ball as anomaly score, inspired by the results in [14] on correlating that distance with an idea of classifier uncertainty. In other words, the score $\mathcal{A}_*$ is written as:

$$\mathcal{A}_*(\mathbf{x}) = -\frac{1}{K} \sum_{k=1}^{K} d_{\mathbb{D}}(\mathbf{0}, \mathbf{x}_k). \tag{5}$$

Finally, we test ensembling the 2 scores. Since the range of $\mathcal{A}$ is $(-\infty, \infty)$ and that of $\mathcal{A}_*$ is $(-\infty, 0]$, we map to $[0, 1]$ using sigmoid and $1 + \tanh$, respectively. Then, using a weight w tuned at validation, the score $\mathcal{A}_{\text{ens}}$ is written as:

$$\mathcal{A}_{\text{ens}}(\mathbf{x}) = (1-w)\times\text{sigmoid}(\mathcal{A}(\mathbf{x}))+w\times(1+\tanh(\mathcal{A}_*(\mathbf{x}))). \tag{6}$$

For each model, we measure for each section s of each machine type m the three area-under-the-ROC-curve (AUC) metrics

[1] github.com/Xiaoccer/MobileFaceNet_Pytorch

[2] github.com/leymir/hyperbolic-image-embeddings

[3] github.com/geoopt/geoopt

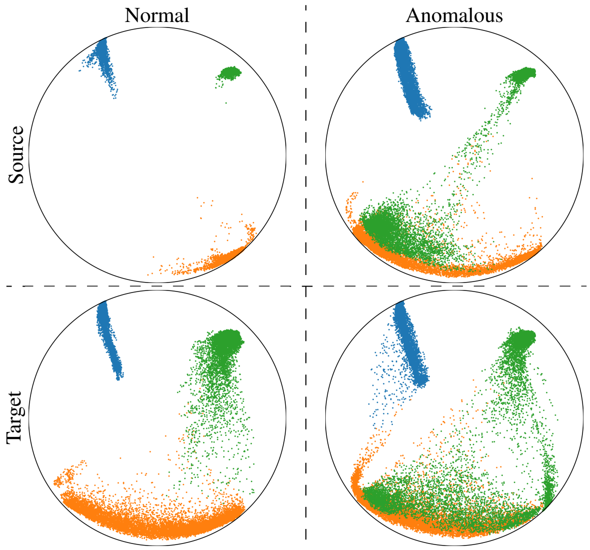

Figure 3: Distribution of the block-level embeddings of the *fan* validation set files in the 2-D Poincaré ball $\mathbb{D}^2$ for the best 2-D hyperbolic model. Each section corresponds to a different color.

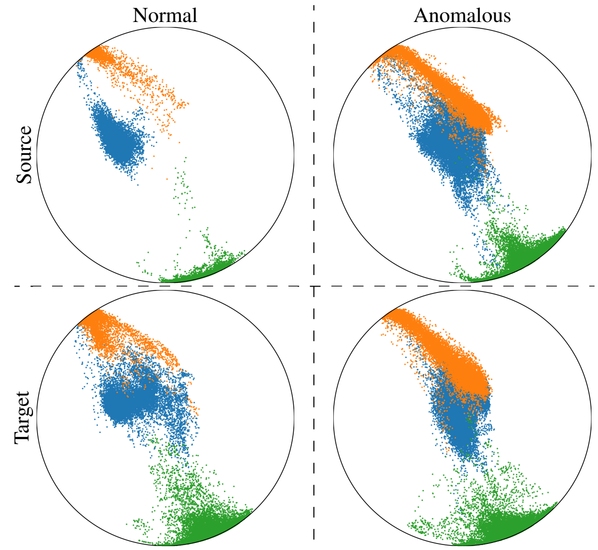

Figure 4: Distribution of the block-level embeddings of the *slider* validation set files.

prescribed for the DCASE 2022 Task 2 challenge [7]: (a) the AUC metric $\mathrm{AUC}^{(S)}_{s,m}$ for the source-domain data, (b) the AUC metric $\mathrm{AUC}^{(T)}_{s,m}$ for the target-domain data, and (c) the pAUC metric $\mathrm{pAUC}_{s,m}$, i.e., the AUC calculated over a low false-positive-rate range of $[0, 0.1]$ for the whole data, which is meant to measure the ability of the system to limit false alarms and be more trustworthy. From those, we compile aggregate metrics $\mathcal{S}_m$ for each machine type m, aggregates $\mathcal{S}^{(S)}_{\mathrm{AUC}}$ and $\mathcal{S}^{(T)}_{\mathrm{AUC}}$ for (resp.) source and target files (also referred to as "AUC (S)" and "AUC (T)"), an aggregate $\mathcal{S}_{\mathrm{pAUC}}$ (also referred to as "pAUC"), and an aggregate overall metric $\mathcal{S}_{\mathrm{ovl}}$ (also referred to as "Overall"), defined as

$$\mathcal{S}_m = \mathcal{H}_s\left\{\mathrm{AUC}^{(S)}_{s,m}, \mathrm{AUC}^{(T)}_{s,m}, \mathrm{pAUC}_{s,m},\ \forall s \text{ of } m \text{ only}\right\} \quad (7)$$

$$\mathcal{S}^{(S)}_{\mathrm{AUC}} = \mathcal{H}_{s,m}\left\{\mathrm{AUC}^{(S)}_{s,m},\ \forall s \text{ of all } m\right\} \quad (8)$$

$$\mathcal{S}^{(T)}_{\mathrm{AUC}} = \mathcal{H}_{s,m}\left\{\mathrm{AUC}^{(T)}_{s,m},\ \forall s \text{ of all } m\right\} \quad (9)$$

$$\mathcal{S}_{\mathrm{pAUC}} = \mathcal{H}_{s,m}\left\{\mathrm{pAUC}_{s,m},\ \forall s \text{ of all } m\right\} \quad (10)$$

$$\mathcal{S}_{\mathrm{ovl}} = \mathcal{H}_{s,m}\left\{\mathrm{AUC}^{(S)}_{s,m}, \mathrm{AUC}^{(T)}_{s,m}, \mathrm{pAUC}_{s,m},\ \forall s \text{ of all } m\right\} \quad (11)$$

where $\mathcal{H}$ is the harmonic mean over the listed indices. Note that the DCASE 2022 official ranking corresponds to $\mathcal{S}_{\mathrm{ovl}}$[4] [7].

4. RESULTS

Tabs 2–3 report the various metrics for our approach and the baselines. For each machine type m and each row, we report the metrics for the checkpoint (and the weight w for $\mathcal{A}_{\mathrm{ens}}$) which performs the highest in terms of $\mathcal{S}_m$ metric on the validation set. For weight w, we try values among $\{0.0, 0.1, \ldots, 0.9, 1.0\}$. We also report published evaluation results for the DCASE 2022 ST and DCASE 2022 AE baselines [8]. We note that DCASE 2022 ST system is similar to "Euclidean," except that it uses a MobileNetV2 [30] backbone and 64×128 blocks of mel-spectrogram frames as input. We do not report results for $L = 128$ using $\mathcal{A}_*$ and $\mathcal{A}_{\mathrm{ens}}$. Indeed, we find the former to be systematically much worse than using $\mathcal{A}$, so that the latter performs at best identically to using $\mathcal{A}$ alone.

In Tab. 2, the hyperbolic-based systems are the best in aggregate for the 2-D systems ($\mathcal{A}_{\mathrm{ens}}$ is best, $\mathcal{A}$ is 2nd best). Both establish competitive per-machine validation metrics. This supports the idea that hyperbolic representations are beneficial, both in terms of class organization (using $\mathcal{A}$) and uncertainty encoding (using $\mathcal{A}_{\mathrm{ens}}$), though uncertainty alone is insufficient (using $\mathcal{A}_*$). These relative strengths appear to carry over well to evaluation based on Tab. 3. Using $\mathcal{A}_{\mathrm{ens}}$ in 2-D results ultimately results in the best evaluation $\mathcal{S}_{\mathrm{ovl}}$ across all conditions. For 128-D systems, the hyperbolic-based system also performs well at validation. The margin is however smaller compared to Euclidean-based systems and it generalizes less well at evaluation. Additionally, we see that the hyperbolic-based systems generalizes better at evaluation on the source domain ($\mathcal{S}^{(S)}_{\mathrm{AUC}}$) but experiences a larger drop on the target domain ($\mathcal{S}^{(T)}_{\mathrm{AUC}}$). We leave further studying of domain generalization to future work.

Further intuition can be gained from observing the distributions of block-level embeddings in Figs. 3–4 for the best 2-D hyperbolic-based system using $\mathcal{A}_{\mathrm{ens}}$ (see Tab. 2). We see how normal source-domain (and, to a lesser extent, target-domain) embeddings cluster much more around the edges of $\mathbb{D}^2$. Meanwhile, anomalous embeddings show a broader footprint, with many more located near the origin. This seems consistent with the aforementioned relationship of distance from the origin as an indicator of classification certainty.

5. CONCLUSION

We explored the use of hyperbolic embeddings for unsupervised anomalous sound detection, which performed favorably compared to Euclidean and ArcFace embeddings. The improvements were most pronounced for small embedding dimensions, which is particularly important for industrial applications when computing resources are limited. In the future, we plan to further explore methods leveraging hyperbolic embeddings. In particular, we plan to investigate how to effectively integrate hyperbolic embeddings into autoencoder-based anomalous sound detection systems.

[4] github.com/Kota-Dohi/dcase2022_evaluator

6. REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.

[2] Y. Kubota, Y. Jiaxing, M. Iwata, M. Murakawa, and T. Higuchi, "Defect detection for RC slab based on hammering echo acoustic analysis," in *Proc. US-Japan Bridg. Eng. Workshop*, 2014.

[3] J. Ye, M. Iwata, K. Takumi, M. Murakawa, H. Tetsuya, Y. Kubota, T. Yui, and K. Mori, "Statistical impact-echo analysis based on grassmann manifold learning: Its preliminary results for concrete condition assessment," in *Proc. Eur. Workshop Struct. Health Monit.*, 2014.

[4] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, "Optimizing acoustic feature extractor for anomalous sound detection based on Neyman-Pearson lemma," in *Proc. EUSIPCO*, 2017.

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

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

[7] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," in *Proc. DCASE*, 2022.

[8] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "First-shot anomaly detection for machine condition monitoring: A domain generalization baseline," *arXiv preprint arXiv:2303.00455*, 2023.

[9] S. Venkatesh, G. Wichern, A. S. Subramanian, and J. Le Roux, "Disentangled surrogate task learning for improved domain generalization in unsupervised anomolous sound detection," in *Proc. DCASE*, 2022.

[10] Y. Zeng, H. Liu, L. Xu, Y. Zhou, and L. Gan, "Robust anomaly sound detection framework for machine condition monitoring," in *Proc. DCASE*, 2022.

[11] I. Kuroyanagi, T. Hayashi, K. Takeda, and T. Toda, "Two-stage anomalous sound detection systems using domain generalization and specialization techniques," in *Proc. DCASE*, 2022.

[12] Y. Deng, J. Liu, and W.-Q. Zhang, "AITHU system for unsupervised anomalous detection of machine working status via sounding," in *Proc. DCASE*, 2022.

[13] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," in *Proc. NeurIPS*, 2018.

[14] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proc. CVPR*, 2020.

[15] W. Chen, X. Han, Y. Lin, H. Zhao, Z. Liu, P. Li, M. Sun, and J. Zhou, "Fully hyperbolic neural networks," in *Proc. Annu. Meet. Assoc. Computat. Linguistics*, 2022.

[16] Y. Guo, X. Wang, Y. Chen, and S. X. Yu, "Clipped hyperbolic classifiers are super-hyperbolic classifiers," in *Proc. CVPR*, 2022.

[17] Q. Liu, M. Nickel, and D. Kiela, "Hyperbolic graph neural networks," *Proc. NeurIPS*, pp. 8228–8239, 2019.

[18] L. Sun, Z. Zhang, J. Zhang, F. Wang, H. Peng, S. Su, and S. Y. Philip, "Hyperbolic variational graph neural network for modeling dynamic graphs," in *Proc. AAAI*, 2021.

[19] A. Jati, N. Kumar, R. Chen, and P. Georgiou, "Hierarchy-aware loss function on a tree structured label space for audio event detection," in *Proc. ICASSP*, 2019.

[20] E. Manilow, G. Wichern, and J. Le Roux, "Hierarchical musical instrument separation," in *Proc. ISMIR*, 2020.

[21] H. Flores Garcia, A. Aguilar, E. Manilow, and B. Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," in *Proc. ISMIR*, 2021.

[22] J. Lee, K. Sung-Bin, S. Kang, and T.-H. Oh, "Lightweight speaker recognition in poincaré spaces," *IEEE Signal Process. Lett.*, vol. 29, pp. 224–228, 2022.

[23] D. Petermann, G. Wichern, A. Subramanian, and J. Le Roux, "Hyperbolic audio source separation," in *Proc. ICASSP*, 2023.

[24] F. Nakashima, T. Nakamura, N. Takamune, S. Fukayama, and H. Saruwatari, "Hyperbolic timbre embedding for musical instrument sound synthesis based on variational autoencoders," in *Proc. APSIPA ASC*, 2022.

[25] R. Shimizu, Y. Mukuta, and T. Harada, "Hyperbolic neural networks++," in *Proc. ICLR*, 2021.

[26] S. Chen, Y. Liu, X. Gao, and Z. Han, "MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices," in *Proc. Chin. Conf. Biom. Recognit.*, 2018.

[27] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. CVPR*, 2018.

[28] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. CVPR*, 2017.

[29] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, 2018.

[31] G. Becigneul and O.-E. Ganea, "Riemannian adaptive optimization methods," in *Proc. ICLR*, 2019.