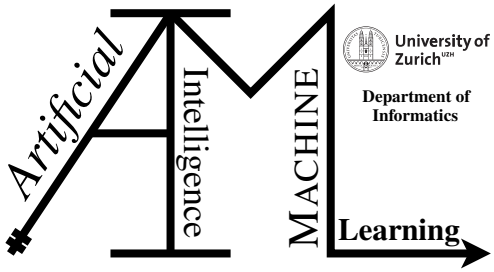
Improved Losses for Open-Set Classification

Master's Thesis

Laurin van den Bergh
16-744-401

Submitted on
August 9 2023

Thesis Supervisor
Prof. Dr. Manuel Günther



Master's Thesis

Author: Laurin van den Bergh, laurin.vandenbergh@uzh.ch

Project period: February 20, 2023 - August 9, 2023

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

I completed this thesis under the supervision of Prof. Dr. Manuel Günther. So first and foremost, I would like to express my gratitude towards him for providing me with the option to write my thesis at the intersection of two very interesting fields, offering his expertise on these topics and advice on how to proceed, as well as giving me the freedom to bring in my own ideas. Also, I would like to thank him for providing me with access to the necessary compute power without which this thesis could not have been conducted. Last but not least, I would like to thank my father and my friends for providing me with emotional support throughout the duration of this thesis, as well as critically questioning my work in a friendly and constructive but challenging way.

Abstract

Open-Set classification (OSC) addresses one of the core issues of traditional classification techniques, namely, the underlying closed-world assumption. The goal of OSC methods is to classify known classes correctly while also rejecting unknown classes. We propose two novel generic loss functions, Margin-OS and Margin-EOS, which combine the Entropic Open-Set and Objectosphere loss with margin-based loss functions used in face recognition tasks, CosFace and ArcFace, to learn discriminative features. We find that the margin has a positive effect on the closed-set accuracy but a mixed effect on the open-set performance. For applications that can tolerate high false positive rates, our losses improve the classification of known classes, but for low false positive rates the margin negatively impacts the training which leads to subpar classification of known samples.

Contents

Chapter 1

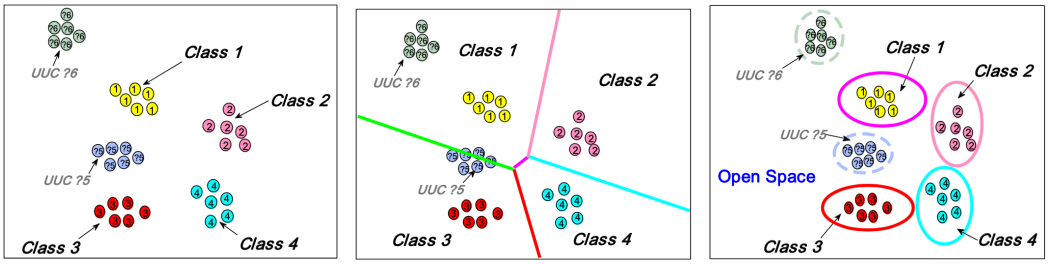
Introduction

Ever since deep convolutional neural networks (DCNNs) such as AlexNet (Krizhevsky et al., 2012) have proved successful on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 (Russakovsky et al., 2015), DCNNs have become dominant approaches for image classification tasks. Classification is the task of categorizing inputs into known classes, *i. e.*, predicting their ground-truth label. The ILSVR challenge, like many others, is a *closed-set classification* task — often simply called classification task — which means that all class labels that occur at test time also have been seen during training (Boult et al., 2019; Wen et al., 2016). As such, this notion of a (closed-set) classification tasks makes a closed-world assumption (Scheirer et al., 2013; Boult et al., 2019; Mahdavi and Carvalho, 2021). It assumes that the world of all possible classes comprises only of the classes contained in the dataset that the algorithm was trained on and consequently a closed-set classifier will classify *any* input as one of the classes seen during training, even if the input visually is completely different to all known inputs.

Neural networks designed for classification can be considered as a composition of two key components: the neural network backbone and the head (see Section 3.1). The backbone takes the preprocessed inputs and transforms them into non-linear *deep features* — or simply features, hence, the backbone is often also called the feature extractor (Goodfellow et al., 2016). Figure 1.1a depicts the deep feature distribution of four known classes 1–4 and two unknown classes 5–6. These deep feature then get fed into the neural network head which performs the actual classification task and returns a probability distribution over all classes. These probabilities are known as softmax probabilities or softmax scores. Finally, label prediction is performed by choosing the class label with the highest softmax score.

The training supervision is traditionally performed via the categorical cross-entropy loss — or softmax loss — which separates the deep features via decision boundaries as depicted in Figure 1.1b. Therefore, the network is encouraged to learn *separable features*, *i. e.* the network is only concerned on which side a sample lies and ignores the distance to the decision boundary of any given sample. These decision boundaries partition the entire deep feature space with respect to the known classes and as such, encompasses the entire open space. For example, it classifies all samples in the top left corner of Figure 1.1b as class 1 even though there lies a different class that is unknown to the classifier, class 6, in the open space behind class 1. If, in this case, a sample of class 6 was given as input, the classifier would predict that it belongs to class 1 as this is the most likely outcome out of all known classes.

Open-Set Classification For real-world applications the closed-world assumption is obviously not justified, since often unwanted or unknown inputs can occur once a model is in an uncontrolled environment (Dhamija et al., 2018). Therefore, we require the classifier to have the option "none of the above" when classifying some input (Dhamija, 2022). In other words, a robust classifier needs to be able to *reject unknown or unwanted inputs* that don't belong to the closed space,



(a) Deep Feature Distribution (b) Closed-set Classification (c) Open-Set Classification

Figure 1.1: DEEP FEATURE SPACE FOR CLASSIFICATION TASKS. *This figure illustrates the deep feature space on which a classification is performed. Deep features for known classes 1–4 and unknown classes 5–6 are shown in (a). Decision boundaries of traditional closed-set classifiers that partition the deep feature space are depicted in (b). Lastly, (c) illustrates the goal of open-set classification, that is preserving and recognizing the open space. Source: (Geng et al., 2021)*

in addition to correctly classifying known classes (Mahdavi and Carvalho, 2021; Dhamija et al., 2018). This requires the classifier to preserve and recognize the open space with the challenge of only having incomplete information about it (Scheirer et al., 2013). Figure 1.1c illustrates this idea by adjusting the decision boundaries to recognize the open space.

(Scheirer et al., 2013) define *open-set recognition (OSR)* as the task, where we want to recognize some classes "in a much larger space of things we do not recognize." (Scheirer et al., 2013) While many use the term *open-set classification (OSC)* synonymously, in this thesis we use the term OSC to refer to tasks or algorithms that perform classification via computing a softmax probability distribution over the known classes. We do so to highlight the fact that all algorithms we experiment with, in this thesis, are extensions to traditional closed-set classification algorithms and to differentiate from classification-based face recognition methods, which do not predict labels based on probability distributions, even when performing classification-like tasks. Accordingly, we consider OSR a broader term that includes OSC as a special case.

According to Dhamija et al. (2018) the goal of OSR methods is twofold:

1. Correctly classify inputs belonging to the known classes.
2. Reject inputs belonging to all other classes.

As such, OSR and in particular OSC aim to improve robustness of AI classification systems against unwanted inputs. This makes OSR a very interesting field for real-world applications and many different approaches from various disciplines exist. In Chapter 2 we introduce two fields which contributed to OSR in ways which are relevant for this thesis, OSC and race recognition.

Problem Formalization Following Dhamija et al. (2018) we can formalize the problem of OSC as follows. We denote the *infinite label space of all classes* as $\mathcal{Y} \subset \mathbb{N}$ which can be further split into two subsets:

- $\mathcal{C} = \{1, \ldots, C\} \subset \mathcal{Y}$ denotes the finite set of *known* classes that a neural network shall learn to classify, for both, closed-set and open-set tasks. Note that the cardinality of this set is $C = |\mathcal{C}| < \infty$.
- $\mathcal{M} = \mathcal{Y} \setminus \mathcal{C}$ denotes the infinite set of *mixed unknown* classes which shall be rejected by a neural network. These are all classes contained in the label space that are not considered

to be known classes and are thus sometimes simply called *unknown* classes. We can further divide $\mathcal{M}$ into the following two subsets:

- $\mathcal{N} \subset \mathcal{M}$ denotes the finite set of mixed unknown classes for which labeled data is available during training. We call the classes contained in $\mathcal{N}$ *negative* classes, but they are sometimes also referred to as *background*, *garbage*, or *known unknown* classes. These classes shall be rejected by the neural network and serve as proxy for $\mathcal{M}$ during training.
- $\mathcal{U} = \mathcal{M} \setminus \mathcal{N}$ denotes the infinite set of *unknown* classes, sometimes also called *unknown unknown* classes. For samples of this class no training samples are available and thus these only occur at test time.

In order to not confuse the mixed unknown and unknown classes, we will not refer to the mixed unknown classes as "unknown" in this thesis. With these notions we can define notations for train and test datasets used in OSC, inspired by the notation from Dhamija et al. (2018). We denote the train dataset of known samples as $\mathcal{D}_{\mathcal{C}}^{\text{train}}$, which contains samples from $\mathcal{C}$, and denote the corresponding test set as $\mathcal{D}_{\mathcal{C}}^{\text{test}}$. Let $\mathcal{D}_{\mathcal{N}}^{\text{train}}$ be the train set of negative samples and $\mathcal{D}_{\mathcal{N}}^{\text{test}}$ be the corresponding test set. Similarly, let $\mathcal{D}_{\mathcal{U}}^{\text{test}}$ be the test set of unknown samples. Accordingly, we denote the test set of mixed unknown samples as $\mathcal{D}_{\mathcal{M}}^{\text{test}} = \mathcal{D}_{\mathcal{U}}^{\text{test}} \cup \mathcal{D}_{\mathcal{N}}^{\text{test}}$. Finally, we denote the full training data by $\mathcal{D}^{\text{train}} = \mathcal{D}_{\mathcal{C}}^{\text{train}} \cup \mathcal{D}_{\mathcal{N}}^{\text{train}}$ and the test data by $\mathcal{D}^{\text{test}} = \mathcal{D}_{\mathcal{C}}^{\text{test}} \cup \mathcal{D}_{\mathcal{M}}^{\text{test}}$, where we denote the size of the training data by $N = \left|\mathcal{D}^{\text{train}}\right|$. Note however, that we use batch processing in all our experiments.

Our Contributions This thesis is motivated by the hypothesis that clustering classes more compactly in the deep feature space enhances a classifiers ability to perform closed-set and open-set classification, with a primary focus on open-set classification. Dhamija (2022) identifies five general approaches of open-set recognition, namely: (1) *learning with unlabeled data*, (2) *deep feature learning via network architecture changes*, (3) *deep feature learning via changing the loss function*, (4) *learning feature distributions*, and (5) *learning with additional data*.

Dhamija (2022) suspects that many advancements in the field of OSR have been made, but accounted as advancements to other fields, *e. g.*, face recognition. Therefore, this thesis aims to explore and improved improve upon OSC loss functions for neural networks by combining recent OSC losses with the fundamental idea used in face recognition methods, namely, learning discriminative features. As such, we explore the combination of the two categories of open-set recognition: *deep feature learning via changing the loss function* and *learning with additional data*.

In particular, we make use of the Entropic Open-Set (EOS) loss and the Objectosphere (OS) loss (Dhamija et al., 2018) for OSC, which incorporate negative samples into the training process to enable effective thresholding of the softmax scores. We combine these with the margin-based loss functions CosFace (Wang et al., 2018b) and ArcFace (Deng et al., 2019), which learn discriminative features through imposing margins in the traditional softmax loss.

We capture these goals with the following research questions:

RQ1: What effect do margins from margin-based loss functions have on the

RQ1a: closed-set performance and

RQ1b: open-set performance

of an OSC task, when trained without negative samples?

RQ2: What effect do margins from margin-based loss functions have on the

RQ2a: closed-set performance and

RQ2b: open-set performance

of an OSC task, when combined with EOS and OS to incorporate negative samples during training?

What We Do and Don't Do We start by exploring how CosFace and ArcFace perform on open-set classification tasks with soft feature normalization (SFN) since recent research suggests that SFN can achieve better performance (Zheng et al., 2018; Liu et al., 2023). We refer to these losses as *SFN-CosFace* and *SFN-ArcFace*. As our main contribution, we propose two novel loss functions *Margin-OS* and *Margin-EOS* that combine the above mentioned approaches in different ways, but both make use of negative samples during training. For all these losses, the normalization of features and weights forces the networks to discriminate between classes only based on the angles of the deep features to each class center (see Section 3.2). This allows us to effectively target the angles with margins to encourage learning discriminative features by clustering deep features for each class more compactly.

In this thesis we only consider methods that perform open-set classification via thresholding softmax probabilities and do not consider any methods that apply thresholds to the logits, *i. e.*, the unnormalized log probabilities, or that introduce a background class (see Section 2.1). For all of the losses considered, we compare hard feature normalization (HFN) and soft feature normalization (SFN) (see Section 3.2.1). However, we do not consider this a focus of this thesis and thus do not formulate it as a research question. Nevertheless, because the feature magnitude plays an important role in margin-based losses and consequently also in our proposed loss functions, we deem it an important consideration (see Section 3.2.2).

We perform preliminary experiments to aid in the process of developing our losses and to visualize deep features, which helps build intuition on how the losses work. Importantly, we consider the four margin-based loss functions SM-Softmax, SphereFace, CosFace, and ArcFace each with a different type of margin. However, due to time restrictions it is impossible for us to consider all four, which is why the preliminary experiments provide empirical guidance on which margin types to prioritize. Then, we conduct our experiments on the ImageNet open-set protocols introduced by Palechor et al. (2023) and evaluate the performance via the open-set classification rate (OSCR) curves introduced by Dhamija et al. (2018) with crucial improvements made by Bisgin et al. (2023). The OSCR curves allow us to evaluate the closed-set performance as a special case when performing an open-set experiment (ses Section 5.2).

Results We find that the margin shows a clear positive effect on the closed-set performance, while the effect on the open-set performance is mixed. The increased closed-set accuracy on methods that are trained with negative samples, is partly explained through the margin, but largely explained through the fact the Margin-OS losses and Margin-EOS losses are forced to discriminate based on the angles only. For losses without negative samples, the effect is almost solely a result of applying a margin.

Our losses can achieve improved open-set classification performance on the known samples when accepting that the classifier will wrongly classify many negative and unknown samples as known (high false positive rate). The higher the applied threshold, the more negative the effect of the margin becomes, as it leads to more known samples being rejected, than for identical networks that do not impose a margin. This leads to underperformance of our margin-based losses on open-set classification tasks where the false positive rate is required to be low. While the size of the margin effects are a bit different when trained with or without negative samples, but qualitatively the observations are similar.

Thesis Outline This thesis is outlined as follows:

- **Related Work**: In this chapter we provide a non-technical overview over relevant approaches from open-set classification and face recognition.
- **Background**: This chapter formally introduces the notation used in this thesis. We motivate the interpretation of the logits via the cosine similarity, which then allows us to formally introduce all loss functions upon which we build our losses.
- **Approach**: In this chapter we introduce and motivate our proposed loss functions Margin-OS and Margin-EOS as well as variations of CosFace and ArcFace that make use of soft feature normalization (SFN), SFN-CosFace and SFN-ArcFace.
- **Experimental Setup**: In this chapter we introduce our toy protocol and the ImageNet open-set protocols on which we conducted our experiments. Furthermore, we introduce the OSCR curve as our evaluation metric and the hyperparameters used in our experiments. Finally, we discuss the network architectures used and the hyperparameter choices.
- **Experiments**: In this chapter we explain the experiments we conducted for answering our research questions and present the results.
- **Discussion**: In this chapter we interpret the results and answer our research questions. Furthermore, we highlight limitations of our work.
- **Conclusion and Future Work**: In this chapter we provide a summary of this thesis and discuss possible areas for future work that is left untouched by this thesis.

Chapter 2

Related Work

In this chapter we provide a non-technical overview over open-set classification approaches and in particular provide an overview into the area of face recognition. Approaches upon which we build our losses are introduced in more detail along with formal definitions in Chapter 3.

2.1 Open-Set Classification

There exist several ways of extending closed-set classification methods to open-set classification methods by providing them with an option to reject inputs (Mahdavi and Carvalho, 2021). Mahdavi and Carvalho (2021) identify two fundamental approaches for addressing open-set classification with approaches that predict probability distributions over known classes: including a background class or thresholding the probability scores.

Background Class Background — or garbage — class approaches make use of negative — or background — samples, that originally belong to various different classes, by relabelling them to form a single background class. The network then treats this like a regular class and learns $C + 1$ classes via softmax loss. If a sample is classified as the background class, it is considered to be unknown. According to Mahdavi and Carvalho (2021) and Dhamija et al. (2018), background class approaches are simple, yet, very effective approaches in practice that aim to learn a separation between the known classes and the background class.

Softmax Score Thresholding Another way of extending a closed-set classifier that predicts a probability distribution over all classes is to apply a probability threshold to the softmax scores. A sample is considered known if its maximal probability score surpasses the threshold, then the label is predicted for which the probability is highest. If the maximal probability does not surpass the threshold for a given sample, then it is considered to be unknown.

Since softmax scores are a "squished" result of the logits and entirely determined by their relative distances, their values can become 1 (up to some precision) when the distances between the logits are high. This makes thresholding them impossible, in which case thresholding the logits directly can lead to better results. However, we only consider softmax thresholding as evaluation method. Furthermore, thresholding softmax scores is susceptible to adversarial or fooling images which achieve high probabilities for images that do not represent any human interpretable class (Nguyen et al., 2015; Goodfellow et al., 2015).

Approaches that aim to threshold softmax scores heavily benefit from learning with negative classes. Dhamija et al. (2018) introduce the Entropic Open-Set (EOS) loss and the Objectosphere

(OS) loss which are trained to achieve a maximal entropy distribution for negative samples, *i. e.*, a uniform distribution over all known classes. This is ideal for applying probability thresholds.

2.2 Face Recognition

Face recognition is an extensively studied computer vision problem and in the context of human biometrics it is also the most used computer vision problem in real-world applications according to Du et al. (2022). We do not provide a comprehensive overview over all possible methods used in face recognition here, but only highlight classification-based face recognition methods. For a broader overview we point to Du et al. (2022). Even though our approaches do not make use of the face recognition pipeline (see Figure 2.1), but use a traditional classification approach, we think it is important to highlight where the idea of imposing a margin comes from and to motivate why the cosine interpretation of the logits is a crucial component of all our proposed losses (see Section 3.2).

The goal of most face recognition tasks is to determine if two images depict the same identity or to identify a person from a list of identities who should be identified, *i. e.*, the gallery. Naturally, most face recognition tasks are evaluated on an open-set protocol, where label prediction is not possible since the the identities from the gallery typically do not overlap with the identities in the training data (Liu et al., 2017). As such, face recognition problems are addressed as transfer learning tasks, where neural networks serve as deep feature extractors (Goodfellow et al., 2016), *i. e.*, they learn face representations from images. This makes face recognition inherently a more open problem than any closed-set classification problem (Scheirer et al., 2013) and therefore requires strong generalization (Scheirer et al., 2013).

2.2.1 End-To-End Face Recognition

Any DCNN-based face recognition pipeline contains the three steps (see Figure 2.1): face detection, face alignment, and face representation (Du et al., 2022). Face representation is broadly considered the core step and is also the only step of interest to us. Note that, due to the fact that face representation is considered the core step of any face recognition task, the term face recognition is sometimes used to refer to face representation.

Figure 2.1 depicts the face recognition process for the face verification task. The face detection step takes an image as input (although other inputs such as videos or a set of images are also possible) and localizes the face region. It typically returns the coordinates of the bounding box (red box in Figure 2.1) as well as a confidence score. The face alignment step takes the detected face and normalizes it to the canonical layout which facilitates the face representation task. The detected faces are, for example, scaled and rotated such that the facial landmarks such as eyes, nose, and corners of the mouth lie on their canonical coordinates within the cropped image. The face representation step then computes deep features that are then used to carry out concrete face recognition tasks.

The two most prominent face recognition tasks are face verification and face identification (Du et al., 2022; Liu et al., 2017). Face verification is the task of deciding if two images depict the same identity by computing the cosine similarity between the feature representations. If the similarity score is above a certain threshold, then the images are considered to depict the same identity, otherwise they are considered to depict different identities. Face identification is the task of identifying some person, via a probe image, from the gallery. Du et al. (2022) note that for "open-set face identification, a prior step is needed, whose target is predicting whether the face belongs to one of the gallery identities or not." As such, open-set face identification can be viewed as a series of face verification tasks, where the probe gets compared to every image on the gallery

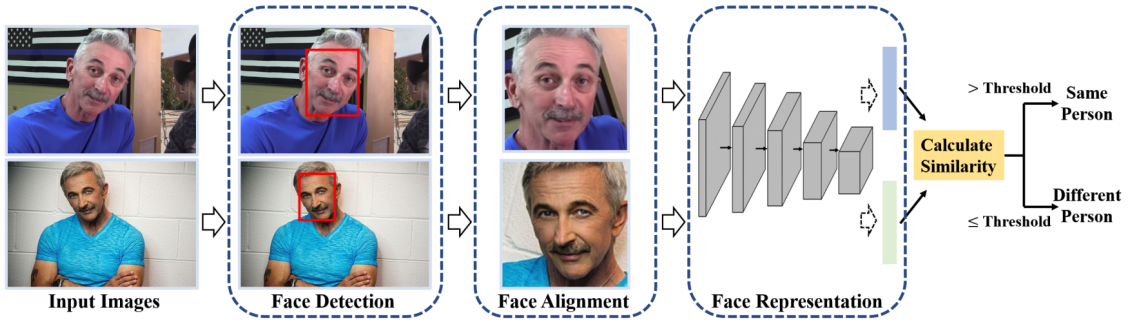


Figure 2.1: END-TO-END FACE RECOGNITION PIPELINE FOR FACE VERIFICATION. *The typical end-to-end face recognition pipeline for face verification consists of three separate steps: face detection, face alignment, and face representation. Face detection takes an input image and localizes the face region. Then, face alignment normalizes the detected face into the canonical layout. Face representation extracts deep feature vectors for each input. Finally, the cosine similarity between two deep feature vectors is computed and compared to a threshold to perform face verification. Source: (Du et al., 2022)*

(Liu et al., 2017). For this reason DCNN-based face recognition is usually interpreted as learning features for face verification in the hope that these generalize well to open-set face identification.

2.2.2 Learning Discriminative Features

Dealing with unseen faces at test time requires strong generalization of the learned features beyond the training data (Wen et al., 2016; Scheirer et al., 2013) and thus the "objective of supervision for any face representation learning is to encourage the faces of same identity to be close and those of different identities to be far apart in the feature space." (Du et al., 2022) Liu et al. (2017) refer to this objective as the *open-set criterion* which states that the maximal intra-class distance between deep features must be smaller than the minimal inter-class distance, given some metric deep feature space. In other words, we want face representations of the same class to be very close in the deep feature space while keeping different classes not only separated but far apart.

Features learned via softmax loss are considered to be separable features and do generally not fulfill the open-set criterion (see Section 3.1). As such, Wen et al. (2016), Liu et al. (2017), Wang et al. (2018b), Deng et al. (2019), and Meng et al. (2021) among many others have highlighted that learning separable deep features — while sufficient for closed-set classification — is insufficient for open-set recognition tasks. For this reason, recent research in face recognition focuses on learning *discriminative features*, which are clustered more compactly in the deep feature space than separable features.

Learning discriminative features can be achieved in various ways with two main contributing factors: the network architecture and the training supervision (Du et al., 2022; Dhamija, 2022). In this thesis we focus on finding new ways of providing supervision for neural networks via different loss functions, so we now only discuss research related to the supervision of face recognition networks and omit discussing specialized network architectures. Most current state-of-the-art approaches, and also the ones relevant to our thesis, are *classification-based methods* Du et al. (2022).

Classification-Based Methods Classification-based face recognition methods consider learning face representations as a multi-class classification task, *i. e.*, the training is conducted via softmax loss. This brings the benefits of being able to leverage the advantages of the softmax loss,

such as scalability to large data sets and number of classes, but comes with the downside that it only learns separable features (Deng et al., 2019).

Classification-based methods are specifically trained to perform transfer learning and are trained as closed-set task on datasets like CASIA-WebFace (Yi et al., 2014), which contains 0.49M face images from 10575 different identities. This is done in the hope that the learned mapping from images to face representations generalizes well to previously unseen face images. Then, the pipeline, as illustrated in Figure 2.1, gets used to carry out face recognition tasks by using the trained network as deep feature extractor for the face representation step. Because the deep features then get compared via the cosine similarity, classification-based methods reinterpret the logits via their cosine interpretation optimize the cosine similarity directly (see Section 3.2). Note that these methods are trained without negative samples, *i. e.*, $\mathcal{D}_{\mathcal{N}}^{\text{train}} = \emptyset$.

A prominent subset of classification-based face recognition losses are the *margin-based losses*. These are fundamental to this thesis and many get introduced in more detail in Section 3.3. To overcome the limitations of separable features learned by the softmax loss, recent methods such as SphereFace (Liu et al., 2017, 2023), SM-Softmax (Liang et al., 2017), CosFace (Wang et al., 2018b), AM-Softmax (Wang et al., 2018a), ArcFace (Deng et al., 2019), and MagFace (Meng et al., 2021) impose a margin in the deep feature space between classes, *i. e.*, identities. This margin artificially penalizes a network during training by computing the softmax activation as if a sample were further away from the true class center than it actually is. By doing so, the margin encourages the network to move the sample closer to the center and further away from the decision boundaries which helps fulfilling the open-set criterion.

Chapter 3

Background

In this chapter we introduce relevant loss functions and fundamental concepts upon which our proposed loss functions build. We start by defining the softmax loss, introducing notation, and terminology. Then, we introduce the margin-based losses SM-Softmax, SphereFace, CosFace, and ArcFace. Lastly, we introduce the OSC approaches Entropic Open-Set loss and Objectosphere loss.

3.1 Softmax Loss

The softmax loss is the backbone of neural network training for many closed-set and open-set classification algorithms. To formally define the softmax loss we need to introduce some notation regarding neural networks. An accompanying overview with a focus on the neural network head and its components is presented in Figure 3.1.

All of the here considered methods are supervised learning techniques which require input-target pairs $(\mathbf{x}_i, \mathbf{t}_i) \in \mathcal{D}^{\text{train}}$ for training, where i is an index over all samples in the data. Notably, we differentiate between the ground-truth labels $\mathbf{y}$ and the targets $\mathbf{t}$ used for training. The labels $\mathbf{y} \in \mathcal{C}^N$ correspond to the known class labels and allow us to keep notation succinct, where N denotes the number of training data points. For example, $\mathbf{y}_1 = 3$ denotes that the ground-truth label of the data point 1 is 3, *i. e.*, $\mathbf{x}_1$ belongs to class 3. The targets $\mathbf{t} \in \mathbb{R}^N$ represent the concrete targets used during training, which can be either equal to the labels but often is a one-hot encoding thereof.

Throughout this thesis we consider a neural network classifier as a composition of two functions: (1) the *backbone* or *deep feature extractor* $\boldsymbol{\phi}_i = B(\mathbf{x}_i) \in \mathbb{R}^K$, which maps a sample $\mathbf{x}_i$ to their deep feature representations $\boldsymbol{\phi}_i$, where $K \in \mathbb{N}$ is the deep feature dimensionality, and (2) the *head* $\mathbf{p}_i = H(\boldsymbol{\phi}_i) \in \mathbb{R}^C$, which maps the deep feature representations $\boldsymbol{\phi}_i$ to a probability distribution $\mathbf{p}_i$ over all known classes, where C is the number of known classes. Since the probability distribution $\mathbf{p}_i$ is the result of a softmax activation, it the probabilities also known as softmax scores. Note that the deep feature representations are also referred to as embeddings. The neural network is then given by the composition $\mathbf{p}_i = H(B(\mathbf{x}_i))$.

Neural Network Head While the feature extractor B performs a crucial task in open-set recognition and open-set classification we do not further discuss it in this thesis. The actual classification task is performed by the head H, which itself is a composition of two layers: (1) a logit layer $\mathbf{z}_i = L(\boldsymbol{\phi}_i) \in \mathbb{R}^C$, where $\mathbf{z}_i$ are the *logits*, and (2) a softmax activation layer $\mathbf{p}_i = \sigma(\mathbf{z}_i) \in \mathbb{R}^C$. The head is then given by:

$$\mathbf{p}_i = H(\boldsymbol{\phi}_i) = \sigma(L(\boldsymbol{\phi}_i)). \tag{3.1}$$

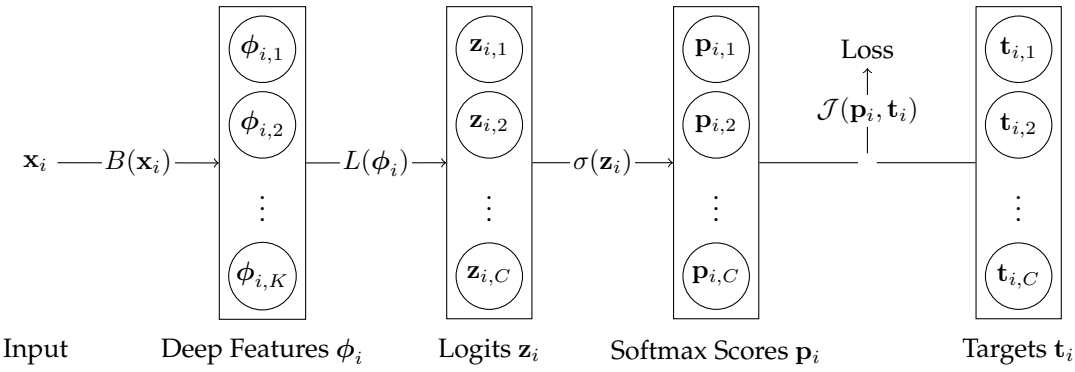


Figure 3.1: SCHEMATIC NEURAL NETWORK OVERVIEW. *This figure illustrates a schematic representation of the general neural network architecture with a focus on the network head. Input samples $\mathbf{x}_i$ get passed through the backbone $B(\mathbf{x}_i)$ which extracts deep features $\boldsymbol{\phi}_i$. The deep features get transformed into logits $\mathbf{z}_i = L(\boldsymbol{\phi}_i)$ via the logit function. Then, the softmax activation $\sigma(\mathbf{z}_i)$ computes softmax scores $\mathbf{p}_i$. Finally, during training, the loss is computed via the loss function $\mathcal{J}(\mathbf{p}_i, \mathbf{t}_i)$ as a function of $\mathbf{p}_i$ and the targets $\mathbf{t}_i$.*

Most neural networks for classification tasks share this basic form of a classifier function, *i. e.*, the head, where the logits are typically defined as a linear transformation of the deep features. The logits for some sample $\mathbf{x}_i$ are then given by:

$$\mathbf{z}_i = L_{\text{linear}}(\boldsymbol{\phi}_i) = \mathbf{W}^\top \boldsymbol{\phi}_i + \mathbf{b}, \tag{3.2}$$

where $\mathbf{W} \in \mathbb{R}^{K \times C}$ is the weight matrix and $\mathbf{b} \in \mathbb{R}^C$ is the bias vector (Goodfellow et al., 2016). For all future discussions we require $\mathbf{b} = \mathbf{0}$ to be the zero vector as this is a crucial assumption for SM-Softmax, SphereFace, CosFace, ArcFace, EOS, and OS. The logits $\mathbf{z}_i$ represent an unnormalized log probability distribution of the sample $\mathbf{x}_i$ over all known classes $c \in \mathcal{C}$, *i. e.*, $\mathbf{z}_{i,c} = \alpha \log P(\mathbf{y}_i = c | \mathbf{x}_i)$ denotes the unnormalized log probability that the true label $\mathbf{y}_i$ for input $\mathbf{x}_i$ is class $c \in \mathcal{C}$, where α is some normalization constant (Goodfellow et al., 2016).

Softmax To obtain normalized probabilities, the logits get passed through the softmax activation function $\sigma(\mathbf{z}_i)$ which outputs a probability distribution over all known classes $c \in \mathcal{C}$. The softmax function is defined element-wise for sample $\mathbf{x}_i$ and class c as follows:

$$\mathbf{p}_{i,c} = \sigma(\mathbf{z}_i)_c = \frac{e^{\mathbf{z}_{i,c}}}{\sum_{c' \in \mathcal{C}} e^{\mathbf{z}_{i,c'}}} \in (0, 1), \qquad \forall c \in \mathcal{C} \tag{3.3}$$

where $c \in \mathcal{C}$ is a class label, $\mathbf{z}_{i,c}$ is the logit value associated with class c, and the full probability distribution is given by $\mathbf{p}_i = (\mathbf{p}_{i,1}, \ldots, \mathbf{p}_{i,C})$ (Goodfellow et al., 2016). The softmax function guarantees that the resulting vector is a probability distribution as it satisfies all required properties. Each probability $\mathbf{p}_{i,c}$ for some sample $\mathbf{x}_i$ and class c is non-negative, *i. e.*, $\mathbf{p}_{i,c} \geq 0$, does not exceed 1, *i. e.*, $\mathbf{p}_{i,c} \leq 1$, and the probabilities sum to 1 over all classes, *i. e.*, $\sum_{c \in \mathcal{C}} \mathbf{p}_{i,c} = 1$. Most importantly, the softmax scores are entirely determined by the differences in the logit values. No differences between the logits, result in a probability distribution where each score is equal to $\frac{1}{C}$, *i. e.* $\mathbf{z}_{i,c} = k,\ \forall c \in \mathcal{C} \Rightarrow \mathbf{p}_{i,c} = \frac{1}{C}$ for some constant k (Dhamija et al., 2018). On the other hand, sufficiently large differences between logits, *e. g.*, one logit $\mathbf{z}_{i,c}$ being far larger than all others, $\mathbf{z}_{i,c'}$ for $c' \neq c$, can result in a score $\mathbf{p}_{i,c} \approx 1$, while all others are around 0, *i. e.* $\mathbf{p}_{i,c'} \approx 0$.

Categorical Cross-entropy Loss The loss function $\mathcal{J}$ measures the "error" — or loss — that a network makes during training and as such guides the supervision. We generally omit stating the explicit arguments for the loss functions and instead highlight here that the arguments include the scores $\mathbf{p}_i$, targets $\mathbf{t}_i$, and optionally the deep features $\boldsymbol{\phi}_i$ for all samples $\mathbf{x}_i$ in the batch. However, we introduce the EOS and softmax loss as a functions of the scores only, since this will simplify upcoming notation.

A neural network for closed-set classification with a softmax activation on the logits is typically trained with the categorical cross-entropy (CCE) loss, which is defined as:

$$\mathcal{J}_{\text{CCE}}(\mathbf{p}_i) = -\sum_{c=1}^{C} \mathbf{t}_{i,c} \cdot \log \mathbf{p}_{i,c} \tag{3.4}$$

$$= -\log \mathbf{p}_{i,\mathbf{y}_i}, \tag{3.5}$$

where $\mathbf{t}_i \in$ is the target vector for sample $\mathbf{x}_i$ and $\mathbf{y}_i$ is the respective ground-truth class label. Note that (3.4) is a more general definition and can be simplified into (3.5) only if $\mathbf{t}_i$ is a one-hot encoded target vector. Since this combination of softmax activation and categorical cross-entropy loss function is very frequently used, it is often often simply called *softmax loss*. Figure 3.2a illustrates the deep feature distributions of a network trained via softmax loss on our toy protocol (see Section 5.1.1).

3.2 Cosine Interpretation of the Logits

Neural networks for face recognition follow a transfer learning approach where the network (backbone and head) is trained via softmax loss and at test time the head gets discarded and only the backbone is used as feature extractor. Concrete face recognition tasks are then conducted by computing cosine similarities between deep feature vectors of identities to determine how similar the deep features are (Liu et al., 2017; Du et al., 2022). For this reason classification-based face representation methods interpret the logits via their cosine to optimize the cosine similarity directly (Wang et al., 2017).

Cosine Similarity The cosine similarity between two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^K$ is defined as:

$$S_{\text{cosine}} := \cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|}, \tag{3.6}$$

where θ is the angle between these two vectors and $\|\cdot\|$ denotes the Euclidean norm. In other words, the cosine similarity between two vectors is the dot product of the normalized vectors. As such, the cosine similarity is neither a distance nor a metric but rather a measure of orientation, which means that it is agnostic to the magnitudes of the vectors and only concerned with their relative orientation around a hypersphere in terms of their angles.

Using the well known fact that the dot product between two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^K$ with angle θ can be written as:

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \, \|\mathbf{v}\| \cos\theta, \tag{3.7}$$

many loss functions interpret the linear logits (3.2), which are simply C dot products (if $\mathbf{b} = \mathbf{0}$), via their cosine interpretation, which allows us to rewrite the logits element-wise as:

$$\mathbf{z}_{i,c} = L_{\text{linear}}(\boldsymbol{\phi}_i)_c = \mathbf{W}_c^\top \boldsymbol{\phi}_i = \|\mathbf{W}_c\| \, \|\boldsymbol{\phi}_i\| \cos(\theta_{i,c}), \tag{3.8}$$

where $\mathbf{z}_{i,c}$ is the logit of sample $\mathbf{x}_i$ corresponding to class c, $\mathbf{W}_c$ is the c-th column of $\mathbf{W}$ (*i. e.*, the class center for class c), and $\theta_{i,c}$ is the angle between $\mathbf{W}_c$ and $\boldsymbol{\phi}_i$.

(a) Softmax Deep Features (b) Normalized Softmax Features (c) Normalized SFN-CosFace Features

Figure 3.2: DEEP FEATURE DISTRIBUTIONS FOR SOFTMAX AND SFN-COSFACE. *This figure shows deep feature distributions of the softmax loss in (a). In, (b) and (c) normalized features for softmax and SFN-CosFace are shown, where the deep features are projected onto a hypersphere, with the respective class centers* $\mathbf{W}_c$ *(lines), which are normalized to fit the scale of the features for illustrative purposes.*

3.2.1 Importance of Normalization

Because cosine similarity is a measure of orientation and thus does not consider the magnitudes of the vectors, we want to normalize the weight and feature vectors in (3.8) to recover the cosine similarity. For this reason classification-based face recognition approaches normalize the weights $\|\mathbf{W}_c\| = 1, \forall c \in \mathcal{C}$ and fix the feature magnitude to a constant $s = \|\boldsymbol{\phi}_i\|$ for all samples $\mathbf{x}_i$ Wang et al. (2017). This results in the logits representing the scaled cosine similarity between the deep feature vector $\boldsymbol{\phi}_i$ and all class centers c and gives us element-wise logits of the form

$$\mathbf{z}_{i,c} = L_{\text{normalized}}(\boldsymbol{\phi}_i)_c = s \cdot \frac{\mathbf{W}_c^\top \boldsymbol{\phi}_i}{\|\mathbf{W}_c\| \, \|\boldsymbol{\phi}_i\|} = s \cdot \cos(\theta_{i,c}), \tag{3.9}$$

where $s \in \mathbb{R}_{>0}$ is a positive scaling constant. The constant s can be interpreted in different ways. For instance, s can be interpreted as a scaling factor on the cosine similarity, to achieve logits of a desired size. Alternatively, it can be interpreted as the normalized deep feature magnitude, where but it defines the radius of the hypersphere on which the deep features get cast.

Figure 3.2b illustrates the deep features of known samples being projected onto a hypersphere for SFN-CosFace which performs feature and weight normalization (see Section 4.1). We choose SFN-CosFace as comparison, since the visualizations show the impact of the margin the clearest. Note that, in contrast to softmax (Figure 3.2b), the class centers for SFN-CosFace are actually "centers" of the class (Figure 3.2c), which is a result of weight and feature normalization.

Normalizing the weights and deep feature magnitudes this forces the network to separate the data in the deep feature space *only via the pairwise angles* (Wang et al., 2017, 2018b; Deng et al., 2019; Liu et al., 2023). These normalizations reduce the classification problem to the problem of separating deep features only via their pairwise angles, which results in the networks distributing the deep features around a hypersphere. Wang et al. (2018b) note that normalizing the feature magnitude learns more discriminative features in the angular space.

Soft vs. Hard Feature Normalization Work by Zheng et al. (2018) and Liu et al. (2023) stress the importance of feature normalization and suggest that *soft feature normalization (SFN)* might be superior to *hard feature normalization (HFN)*, the latter being how the features are normalized in

(3.9) and also how CosFace and ArcFace were originally introduced. Zheng et al. (2018) perform SFN by adding a magnitude regularization term to the softmax loss and SphereFace loss (see Section 4.1) and in both cases achieve significant performance improvements on face recognition tasks on a variety of datasets. They call their SFN approach the Ring loss where s serves as the target feature magnitude that shall be learned by the networks. Liu et al. (2023) find in their revision of the original SphereFace formulation (Liu et al., 2017) that SFN achieves better performance than HFN on datasets with high quality, while HFN performs better on noisy images. While SFN comes at the cost of an additional hyperparameter — the weight associated with the regularization term — that potentially needs tuning, Zheng et al. (2018) find that the improvements over HFN stay significant for a large range of values for the weight.

3.2.2 Importance of the Feature Magnitude

The scaling constant s is of great importance for the training of classification-based networks as well as prediction via softmax scores because it directly influences the probability scores that the network can achieve (Zhang et al., 2019) since the probabilities are determined by the difference between the values of the logits.

The original SphereFace was introduced without feature normalization which lead to unstable training that was critiqued by (Wang et al., 2018b) and Liu et al. (2023) themselves who fixed it to some empirically chosen constant s that provided good performance. Wang et al. (2018b) provide a lower bound on s (see (3.11)) for expected maximal softmax scores and argue that "s should be larger to deal with more classes since the growing number of classes increase the difficulty for classification in the relatively compact space. A hypersphere with large radius s is therefore required for embedding features with small intra-class distance and large inter-class distance." However, they proceed to choose a value for s significantly larger than the any reasonable lower bound they could have achieved, as they empirically set $s = 64$. Deng et al. (2019) copy the value $s = 64$ from Wang et al. (2018b) while Liu et al. (2023) choose a lower value of $s = 30$ and achieve significant improvements.

Intuition and Desired Behavior Zhang et al. (2019) provide a more in-depth analysis of the parameter s for a classifier with logits of the form (3.9), for which we have $\mathbf{z}_{i,c} \in [-s, s]$ since the domain for the angles is $[0, \pi]$. For simplicity we assume that even with SFN the logits lie in $[-s, s]$ even though the range of logits is likely significantly larger. However, Zhang et al. (2019) find that empirically we can further reduce the domain of the angles to $[0, \frac{\pi}{2}]$ since in practice the maximal angle of deep features to any non-ground-truth class is around $\frac{\pi}{2}$, *i. e.* their angle is usually around 90 degrees. This leads to logits $\mathbf{z}_{i,c} \in [0, s]$.

They analyze the probability score of a sample $\mathbf{x}_i$ to some class center c, as a function of the angle between the deep feature and the class center, $\theta_{i,c}$. Of particular interest here are the scores for the ground truth labels, *i. e.*, $\mathbf{p}_{i,\mathbf{y}_i}$. This provides probability curves which can be nicely analyzed, visualized, and interpreted. They are given by:

$$P(\mathbf{y}_i = c|\mathbf{x}_i; \theta_{i,c}) = \frac{e^{s\cdot\cos(\theta_{i,c})}}{e^{s\cdot\cos(\theta_{i,c})} + (C-1)e^{s\cdot 0}}, \tag{3.10}$$

where one can optionally also include any margins, *e. g.*, the additive cosine margin (3.15). These curves are not exact, however, because they rely on two assumptions: (1) HFN of the deep features and (2) the average angle of any known sample to all other class centers is $\frac{\pi}{2}$. We verified empirically that the second assumption is fulfilled for all losses with almost no deviation for any individual test sample. For an visualization of these probability curves see Figure 6.6.

Zhang et al. (2019) argue that we want these curves to *gradually decrease* from 1 to 0 as the angle of the sample to the ground truth class center increases from 0 to $\frac{\pi}{2}$. They find that when s is too

small the probabilities for the true class fail to reach 1 even when $\theta_{i,\mathbf{y}_i} = 0$. This is undesirable as this leads to the network being punished by the loss even though it is as confident as it can be — hence the need for a lower bound. When s is too large — and Zhang et al. (2019) explicitly mention $s = 64$ as being too large — and increasing the angle from 0 to $\frac{\pi}{2}$ (*i. e.* moving further away from the ground truth class center), the probability curve will stay close to 1 until almost $\theta_{i,\mathbf{y}_i} = \frac{\pi}{2}$ before it falls off steeply to 0. This, too, is undesirable since this means that the probabilities stay close to 1 even if the angle $\theta_{i,\mathbf{y}_i}$ approaches $\frac{\pi}{2}$. As such a too large feature magnitude "s may fail to penalize mis-classified samples and cannot effectively update the networks to correct mistakes" because of very small gradients of the probability curves (Zhang et al., 2019). In summary, s should be as large as necessary but as small as possible.

Lower Bound Wang et al. (2018b) provide a lower bound on s as a function of the number of classes C and the expected minimum posterior probability $\hat{\mathbf{p}}_{i,c}$ of a sample $\mathbf{x}_i$ and some class center c. The expected minimum posterior probability is a hyperparameter that represents the lower bound on the expected maximal softmax score for sample $\mathbf{x}_i$ and class c when their angle is 0, *i. e.* $\theta_{i,c} = 0$, and consequently $\cos(\theta_{i,c}) = 1$ holds. The lower bound is given by:

$$s \geq \frac{C-1}{C} \log \frac{(C-1)\hat{\mathbf{p}}_{i,c}}{1-\hat{\mathbf{p}}_{i,c}}, \tag{3.11}$$

where $C > 1$. When s fulfills this bound, then a sample $\mathbf{x}_i$ with $\theta_{i,c} = 0$ achieves in expectation a maximal softmax score $\mathbf{p}_{i,c} \geq \hat{\mathbf{p}}_{i,c}$.

Effectively, this lower bound replaces the hyperparameter s with the hyperparameter $\hat{\mathbf{p}}_{i,c}$ for which it is much simpler to find a suitable value, since we have a clear understanding and interpretation for it. We would like $\hat{\mathbf{p}}_{i,c}$ to be very close to 1 but potentially a bit smaller to guarantee that on average the maximum softmax scores for the known classes are larger than $\hat{\mathbf{p}}_{i,c}$.

Note that this lower bound does not take into account the additive cosine margin nor the additive angular margin. During training, this will result in $\cos(\theta_{i,\mathbf{y}_i}) - m = 1 - m < 1$ or $\cos(\theta_{i,\mathbf{y}_i} + m) = \cos(m) < 1$ for the true class label $\mathbf{y}_i$ and $m \in (0, \pi)$, which may lead to wrongful penalization of the network even though network learned to classify the sample perfectly.

3.3 Margin-Based Losses

We now formally introduce the margin-based losses that we consider in this thesis: SM-Softmax, SphereFace, CosFace, and ArcFace, each of which applies a different type of margin. The margin-based losses benefit strongly from having feature and weight normalization, because forcing the networks to discriminate only based on the angle is critical for imposing margins on the angles, as it prevents the networks from circumventing the margin (Wang et al., 2018b). Which leads to classes being clustered more compactly with larger differences between samples from different classes. This is illustrated in Figure 3.2, where Figure 3.2b depicts traditional softmax and Figure 3.2c depicts our SFN-CosFace loss which imposes a margin between known classes (see Section 4.1).

Highlighting the fact that these losses are essentially just modifications in the logits trained with standard softmax loss (3.5), they can all be written very succinctly as:

$$\mathcal{J}_{\mathrm{T}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{J}_{\mathrm{CCE}} \left(\sigma(L_{\mathrm{T}}(\boldsymbol{\phi}_i)) \right) \tag{3.12}$$

where T is the type of logits used, *e. g.*, T = SM-Softmax for SM-Softmax logits (3.13).

3.3.1 SM-Softmax Loss - Additive Logit Margin

SM-Softmax is perhaps the most straight forward extension of the softmax loss. Liang et al. (2017) don't explicitly mention if their loss is designed for face recognition tasks and conduct their experiments on the CIFAR-10 and CIFAR-100 datasets, both are designed for general image classification tasks, but they base their approach on L-Softmax (Liu et al., 2016) which is designed for face recognition tasks and is similar to SphereFace. They do, however, explicitly follow the goal of learning discriminative features by optimizing the open-set criterion which makes the SM-Softmax a noteworthy consideration for this thesis.

SM-Softmax does neither use feature normalization nor weight normalization and targets the logits directly with its margin. As such, we call it an additive logit margin m_{logit} and the logit for sample $\mathbf{x}_i$ and class c is given by:

$$\mathbf{z}_{i,c} = L_{\text{SM-Softmax}}(\boldsymbol{\phi}_i)_c = \mathbf{W}_c^\top \boldsymbol{\phi}_i - \delta_{c\mathbf{y}_i} \cdot m_{\text{logit}}, \tag{3.13}$$

where $\mathbf{y}_i$ denotes the ground-truth label and $\delta_{c\mathbf{y}_i}$ is the Kronecker delta which equals 1 if $c = \mathbf{y}_i$ and 0 otherwise. Liang et al. (2017) do not provide any bounds on m_{logit}, but explore values $m_{\text{logit}}[0.1, 0.9]$ and find that values around 0.3 provide the best results. Trivially, however, any reasonable margin must be a positive real valued number, *i. e.* $m_{\text{logit}} \in \mathbb{R}_{>0}$.

3.3.2 SphereFace Loss - Multiplicative Angular Margin

SphereFace is designed to be an extension of the softmax loss that imposes a multiplicative margin on the angle of a sample to its ground-truth class center. Liu et al. (2017) call their proposed loss the A-Softmax loss, however, for clarity we refer to it as SphereFace loss. Unlike SM-Softmax, SphereFace interprets the logits via their cosine and normalizes the weights $\mathbf{W}_c = 1,\ \forall c \in \mathcal{C}$. SphereFace was originally introduced without feature normalization (Liu et al., 2017) which lead to instabilities during training. As such, we introduce SphereFace with HFN here, following Liu et al. (2023). For sample $\mathbf{x}_i$ and class c, SphereFace has logits of the form:

$$\mathbf{z}_{i,c} = L_{\text{SphereFace}}(\boldsymbol{\phi}_i)_c = \begin{cases} s \cdot \left((-1)^{k_i} \cos(m_{\text{mult}} \theta_{i,\mathbf{y}_i}) - 2k_i\right) & \text{if } c = \mathbf{y}_i \\ s \cdot \cos(\theta_{i,c}) & otherwise \end{cases}, \tag{3.14}$$

where $m_{\text{mult}} \in \mathbb{R}_{>1}$ denotes the multiplicative angular margin and $k_i \in \mathbb{N}$ is chosen such that $\theta_{i,\mathbf{y}_i} \in \left[\frac{k_i \pi}{m_{\text{mult}}}, \frac{(k_i+1)\pi}{m_{\text{mult}}}\right]$ holds. Liu et al. (2017) use $(-1)^{k_i} \cos(m_{\text{mult}} \theta_{i,\mathbf{y}_i}) - 2k_i$ as an approximation for $\cos(m_{\text{mult}} \theta_{i,\mathbf{y}_i})$ because this allows them to get around the restriction that $\theta_{i,\mathbf{y}_i} \in [0, \frac{\pi}{m_{\text{mult}}}]$ must hold. In the code accompanying their paper, Liu et al. (2023) set $k_i = \lfloor \frac{m_{\text{mult}} \theta_{i,\mathbf{y}_i}}{\pi} \rfloor$.

3.3.3 CosFace Loss - Additive Cosine Margin

CosFace extends the logits (3.9) by imposing an additive margin $m_{\text{cos}} \in \mathbb{R}_{>0}$ on the cosine of the ground-truth class (Wang et al., 2018b) and as such the logits are linear in the margin. It is designed to directly target the unscaled cosine similarities, since it is applied before multiplying by s. This can also be interpreted as a margin on the logit of the true class (Du et al., 2022), where now the margin is scaled to $s \cdot m_{\text{cos}}$. As Zhang et al. (2019) demonstrate we can succinctly express the CosFace loss via their logits:

$$\mathbf{z}_{i,c} = L_{\text{CosFace}}(\boldsymbol{\phi}_i)_c = s \cdot (\cos(\theta_{i,c}) - \delta_{c\mathbf{y}_i} \cdot m_{\text{cos}}), \tag{3.15}$$

where m_{cos} is the additive cosine margin and $\delta_{c\mathbf{y}_i}$ is the Kronecker delta.

During training, the cosine margin decreases the value of the cosine for the logit corresponding to the ground-truth class and thus indirectly simulates points being further away in terms of their angle to the true class. Wang et al. (2018b) refer to the resulting loss, *i. e.*, softmax loss in conjunction with the CosFace logits, as Large Margin Cosine Loss (LMCL).

3.3.4 ArcFace Loss - Additive Angular Margin

ArcFace extends the logits (3.9) by imposing an additive margin $m_{\text{ang}} \in \mathbb{R}_{>0}$ on the angle towards the true class center (Deng et al., 2019). In contrast to CosFace, ArcFace targets the deep feature space directly by imposing the margin on the angle. This results in a margin that non-linearly impacts the logits but allows for simple interpretation of the clustering in the deep feature space. For this thesis we restrict the discussion to the standard ArcFace version and do not consider sub-center ArcFace as this addresses issues particular to face recognition which are irrelevant for our application. Following Zhang et al. (2019) we can succinctly express ArcFace via the logits:

$$\mathbf{z}_{i,c} = L_{\text{ArcFace}}(\boldsymbol{\phi}_i)_c = s \cdot \cos(\underbrace{\theta_{i,c} + \delta_{c\mathbf{y}_i} \cdot m_{\text{ang}}}_{\tilde{\theta}_{i,c}}), \tag{3.16}$$

where m_{ang} is the additive angular margin and $\delta_{c\mathbf{y}_i}$ is the Kronecker delta. Following the implementation from Liu et al. (2023)[1] we clamp the modified angle values $\tilde{\theta}_{i,c}$ to the interval $[0, \pi]$ to avoid falsely punishing the network in case the angle $\theta_{i,c}$ is smaller than the margin m_{ang}, in which case the network would learn to decrease the angle by turning it negative and trying to learn $-m_{\text{ang}}$ as optimal angle. Which it could not achieve, since the angles are always considered to be positive, when measuring the cosine similarity.

3.4 OSC Losses

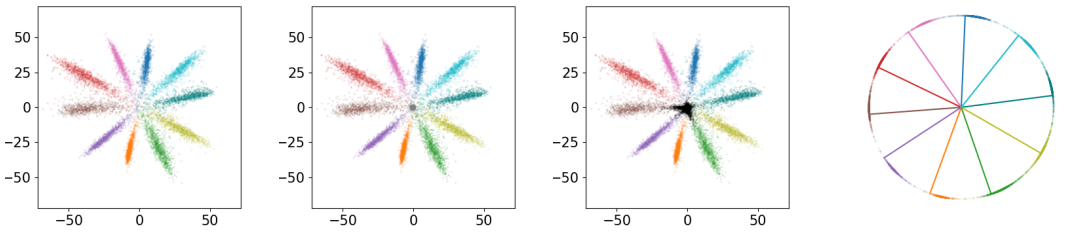
Dhamija et al. (2018) propose two loss functions for training neural networks for open-set classification with negative samples. Both approaches, Entropic Open-Set Loss (EOS) and Objectosphere Loss (OS) — the latter being an extension of the former — aim to learn the networks in such a way that for unknown samples it returns a uniform distribution, *i. e.*, every class being equally likely. This lends itself well to applying probability thresholds to the softmax scores.

3.4.1 Entropic Open-Set Loss

The Entropic Open-Set loss can be viewed as a generalization of the softmax loss, because it extends the loss to deal with negative samples. The core idea of EOS is to train the network such that for any unknown sample $\mathbf{x}_i \in \mathcal{D}_{\mathcal{N}}^{\text{train}}$ the resulting probability distribution $\mathbf{p}_i = H(B(\mathbf{x}_i))$ has maximal entropy (Shannon, 1948), *i. e.*, it is a uniform distribution. We deviate from our naming convention for loss functions here and define the EOS loss $\mathcal{J}_{\text{EOS}}$ for a single as a function of $\mathbf{p}_i$, which will simplify upcoming notation:

$$\mathcal{J}_{\text{EOS}}(\mathbf{p}_i) = \begin{cases} -\log \mathbf{p}_{i,\mathbf{y}_i} & \text{if } \mathbf{x}_i \in \mathcal{D}_{\mathcal{C}}^{\text{train}} \\ -\frac{1}{C}\sum_{c=1}^{C} \mathbf{p}_{i,c} & \text{if } \mathbf{x}_i \in \mathcal{D}_{\mathcal{N}}^{\text{train}} \end{cases}, \tag{3.17}$$

[1] Source code is available at: https://github.com/ydwen/opensphere/blob/main/model/head/arcface.py



(a) Known (b) Negatives (c) Unknowns (d) Normalized Features

Figure 3.3: DEEP FEATURE DISTRIBUTIONS FOR THE EOS LOSS. *EOS deep feature distributions of the 10 known classes are shown in (a) as learned by the EOS loss . In, (b) and (c) the deep features for negative samples (gray) and unknown samples (black) are superimposed, respectively. Finally, (d) shows normalized features, i. e., being projected onto a hypersphere, with the respective class centers* $\mathbf{W}_c$ *(lines), which are normalized to fit the scale of the features for illustrative purposes.*

where $\mathbf{p}_{i,c} = H(B(\mathbf{x}_i))_c$ is the softmax score for class c associated with sample $\mathbf{x}_i$. In practice, we can implement the EOS loss via the general CCE loss (3.4) by setting the targets for negative samples j to $\mathbf{t}_{j,c} = \frac{1}{C}$, $\forall c \in \mathcal{C}$, which gives us the equivalent expression:

$$\mathcal{J}_{\text{EOS}}(\mathbf{p}_i) = \sum_{c=1}^{C} \mathbf{t}_{i,c} \cdot \log \mathbf{p}_{i,c} \tag{3.18}$$

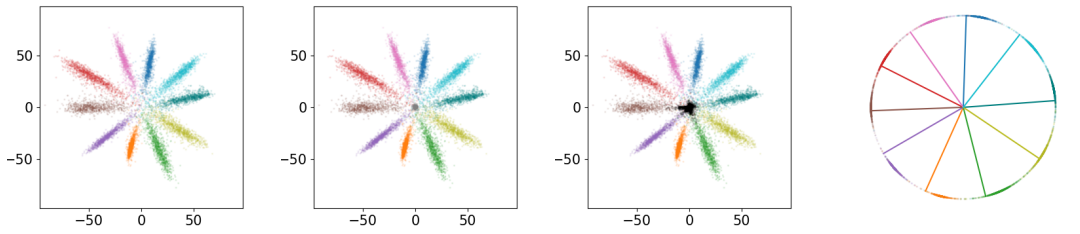
This requires the implementation to use one-hot encoded targets instead of labels.

Dhamija et al. (2018) show that for negative samples, *i. e.* $\mathbf{x}_i \in \mathcal{D}_{\mathcal{N}}^{\text{train}}$, $\mathcal{J}_{\text{EOS}}$ is minimized if the logit values for all classes are equal. However, such a minimum is not unique. Figure 3.3 illustrates the deep features of learned by the EOS loss on our toy protocol. We superimpose the negative and unknown test samples in Figure 3.3b and Figure 3.3c, respectively, which have a tendency to be drawn to the origin of the deep feature space.

3.4.2 Objectosphere Loss

In order to address the issue of a non-unique minimum, Dhamija et al. (2018) refine the EOS loss with a regularization term that guarantees that the loss for a negative sample is minimized only for the zero vector in the deep feature space. They call the resulting loss the Objectosphere loss, which we call OS loss, for short. As such, the OS loss exploits the natural tendency of negative and unknown samples being drawn to the origin of the deep feature space when training with softmax loss or EOS loss (Dhamija et al., 2018). This has the desirable property that logits with a feature magnitude of zero have a logit vector that is the zero vector, which yields a softmax distribution with maximal entropy, *i. e.* $\|\boldsymbol{\phi}_i\| = 0 \Rightarrow \mathbf{z}_{i,c} = 0,\ \forall c \in \mathcal{C} \Rightarrow \mathbf{p}_{i,c} = \frac{1}{C}$.

While Dhamija et al. (2018) did not name the regularization term, we refer to it as *objectosphere regularizer* — or OS-regularizer, for short. The OS-regularizer targets the deep features directly and draws negative samples towards the origin of the deep feature space while casting known samples outside of a sphere to differentiate the knowns and negatives based on their feature magnitude. Figure 3.4 illustrates the deep features learned via the OS loss, along with the negative and unknown test samples. We also depict the normalized deep features of the known samples in Figure 3.4d.



(a) Known (b) Negatives (c) Unknowns (d) Normalized Features

Figure 3.4: DEEP FEATURE DISTRIBUTIONS FOR THE OS LOSS. *OS deep feature distributions of the 10 known classes are shown in (a) as learned by the OS loss . In, (b) and (c) the deep features for negative samples (gray) and unknown samples (black) are superimposed, respectively. Finally, (d) shows normalized features, i. e., being projected onto a hypersphere, with the respective class centers* $\mathbf{W}_c$ *(lines), which are normalized to fit the scale of the features for illustrative purposes.*

The objectosphere loss $\mathcal{J}_{OS}$ is defined as:

$$\mathcal{J}_{\text{OS}} = \sum_{i=1}^{N} \mathcal{J}_{\text{EOS}}(\mathbf{p}_i) + \lambda \cdot \begin{cases} \max(0, \xi - \|\boldsymbol{\phi}_i\|)^2 & \text{if } \mathbf{x}_i \in \mathcal{D}_{\mathcal{C}}^{\text{train}} \\ \|\boldsymbol{\phi}_i\|^2 & \text{if } \mathbf{x}_i \in \mathcal{D}_{\mathcal{N}}^{\text{train}} \end{cases}, \tag{3.19}$$

where $\lambda \in \mathbb{R}_{>0}$ is a hyperparameter controlling the weight of the regularization term and ξ is the radius of the sphere. The hyperparameter ξ can be interpreted as the minimally accepted feature magnitude for known samples.

Chapter 4

Approach

In this chapter we introduce three generic loss functions SFN-Margin loss, Margin-OS loss, and Margin-EOS loss with fundamentally different approaches to the open-set classification problem. We consider these to be "generic loss functions" since they can theoretically use various different types of margins by simply choosing different logits. We consider the additive cosine margin and the additive angular margin, as introduced by Wang et al. (2018b) and Deng et al. (2019), respectively, as these have shown the most promising results in our preliminary testing (see Section 6.1.1).

For each of these generic loss functions we introduce a zero-margin version that is identical to the main losses but sets the margin to 0. This allows us to *study the effect of the respective margins in isolation* by serving as a baseline to which we can compare the respective cosine and angular margin versions and correct for effects caused by normalizing weights and features. Notably, however, these zero-margin versions are not the benckmarks to which we compare the classification performance (see Section 6.2). For SFN-Margin and Margin-OS we can provide visualizations of the deep feature distributions on our toy protocol (see Section 6.1.4). Unfortunately, we cannot provide deep feature visualizations for Margin-EOS because it learns orthogonal class centers, which cannot be visualized in 2D space.

4.1 SFN-Margin Losses

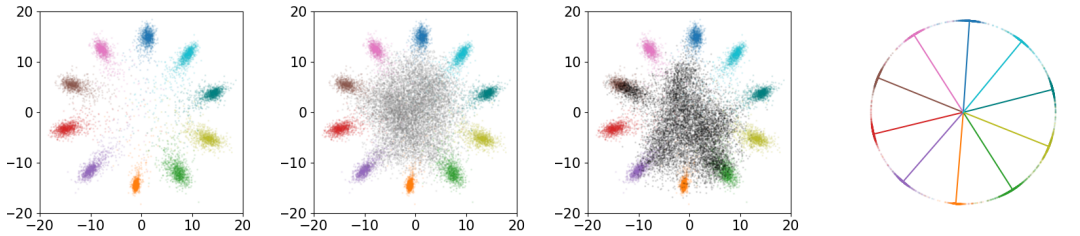
To address research question RQ1 we adapt the margin-based face recognition loss functions CosFace and ArcFace to be used for open-set classification tasks, *i. e.*, for performing label prediction at test time via softmax scores. We train these networks without negative samples, *i. e.* $\mathcal{D}_{\mathcal{N}}^{\text{train}} = \emptyset$.

CosFace and ArcFace were originally introduced with HFN and a scaling parameter s. However, newer research suggests that SFN achieves superior performance (Zheng et al., 2018; Liu et al., 2023) and we can confirm these findings for CosFace and ArcFace with our preliminary experiments (see Section 6.1.2). Thus, we define *SFN-CosFace* and *SFN-ArcFace* — analogously to work by Zheng et al. (2018) — by adding the Ring loss and keeping the feature magnitude variable in the logits. This gives us SFN-CosFace logits for sample $\mathbf{x}_i$ and class c of the form

$$L_{\text{SFN-CosFace}}(\boldsymbol{\phi}_i)_c = \|\boldsymbol{\phi}_i\| \cdot (\cos(\theta_{i,c}) - \delta_{c\mathbf{y}_i} \cdot m_{\cos}) \tag{4.1}$$

and analogously SFN-ArcFace logits of the form

$$L_{\text{SFN-ArcFace}}(\boldsymbol{\phi}_i)_c = \|\boldsymbol{\phi}_i\| \cdot \cos(\underbrace{\theta_{i,c} + \delta_{c\mathbf{y}_i} \cdot m_{\text{ang}}}_{\tilde{\theta}_{i,c}}), \tag{4.2}$$



(a) Known (b) Negatives (c) Unknowns (d) Normalized Features

Figure 4.1: DEEP FEATURE DISTRIBUTIONS FOR THE SFN-ARCFACE LOSS. *SFN-ArcFace deep feature distributions of the 10 known classes are shown in (a) as learned by the SFN-ArcFace loss . In, (b) and (c) the deep features for negative samples (gray) and unknown samples (black) are superimposed, respectively. Finally, (d) shows normalized features, i. e., being projected onto a hypersphere, with the respective class centers* $\mathbf{W}_c$ *(lines), which are normalized to fit the scale of the features for illustrative purposes.*

where $\delta_{c\mathbf{y}_i}$ is the Kronecker delta. Like for ArcFace, the modified angle $\tilde{\theta}_{i,c}$ is clamped to the interval $[0, \pi]$ to avoid learning negative angles. At test time the margins are set to 0.

This is a combination of CosFace and ArcFace respectively with the Ring loss (Zheng et al., 2018) with the only difference that we do not multiply the regularization term by $\frac{1}{2}$, which does not change its behavior, but means that values of λ should be compared cautiously. This gives us the SFN-CosFace and SFN-ArcFace loss:

$$\mathcal{J}_{\mathrm{T}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{J}_{\mathrm{CCE}}\left(\sigma(L_{\mathrm{T}}(\boldsymbol{\phi}_i))\right) + \lambda \left\| s - \|\boldsymbol{\phi}_i\| \right\|^2, \tag{4.3}$$

for $\mathrm{T} \in \{\text{SFN-CosFace}, \text{SFN-ArcFace}\}$, where λ is the weight of the regularization term and s is the target deep feature magnitude.

We believe this to be a very straight forward extension of classification-based loss functions used in face recognition for typical classification tasks and thus do not consider this an inherently novel approach, however, this builds a relevant basis for our proposed methods Margin-OS (Section 4.2) and lets us evaluate whether the inclusion of a margin is helpful for open-set classification. To evaluate the impact of the margin independently from the weight and feature normalization we also consider an additional loss that is identical to (4.3) but sets the margin to 0 during training. We call this loss *SFN-Norm* since the only factor differentiating it from softmax loss (apart from the Ring loss) is the normalization of the logit weights, *i. e.* $\|\mathbf{W}_c\| = 1,\ \forall c \in \mathcal{C}$ and the SFN of the deep features. Figure 4.1 illustrates the learned deep features for the SFN-ArcFace loss, with superimposed negative and unknown test samples. Additionally, we depict the deep features projected onto a sphere with normalized class centers in Figure 4.1d to highlight the impact of imposing a margin. We can see that the known classes are oriented around the sphere with the negative and unknown samples learning generally smaller deep feature magnitudes and being distributed relatively equally across the deep feature space.

4.2 Margin-OS Losses

Margin-OS is our first proposed generic loss to address research question RQ2. It builds directly on the loss functions (4.3) and extends them to train with negative samples in a similar fashion

to the OS loss, as it tries to explicitly encourage negative samples being drawn to the origin in comparison to the SFN-Margin losses. Similarly to the OS loss, this results in uniform probability scores for negative and unknown samples.

Since the OS-regularizer and the Ring loss share similarities in how they address the deep feature magnitudes directly and the OS-regularizer already handles negative samples in an effective and intuitive way, it is perhaps natural and straight forward to replace the Ring loss with the OS-regularizer. One problem with this idea is that the feature magnitudes of known samples are not bound above for the OS-regularizer which can run into the issues discussed in Section 3.2.2. While this is no issue in for EOS and OS — which keep the logit weights unnormalized which can counteract these problems — this is a potential problem for Margin-OS since it builds on the SFN-Margin losses (4.3) which keep the weights normalized. We adapt the OS-regularization term to symmetrically penalize deviations of the feature magnitudes from the target feature magnitudes s of known samples, we call it the *symmetric OS-regularizer* (see Section 6.1.3). In accordance with the namesakes of the margin losses CosFace and ArcFace as well of the OS-regularizer, we call these losses *Cos-OS* and *Arc-OS* respectively. The Cos-OS loss is given by:

$$\mathcal{J}_{\text{Cos-OS}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{J}_{\text{CCE}}\left(\sigma(L_{\text{SFN-CosFace}}(\boldsymbol{\phi}_i))\right) + \lambda \begin{cases} \left\|s - \|\boldsymbol{\phi}_i\|\right\|^2 & \text{if } \mathbf{x}_i \in \mathcal{D}_{\mathcal{C}}^{\text{train}} \\ \|\boldsymbol{\phi}_i\|^2 & \text{if } \mathbf{x}_i \in \mathcal{D}_{\mathcal{N}}^{\text{train}} \end{cases} \tag{4.4}$$

and the Arc-OS loss is given by

$$\mathcal{J}_{\text{Arc-OS}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{J}_{\text{CCE}}\left(\sigma(L_{\text{SFN-ArcFace}}(\boldsymbol{\phi}_i))\right) + \lambda \begin{cases} \left\|s - \|\boldsymbol{\phi}_i\|\right\|^2 & \text{if } \mathbf{x}_i \in \mathcal{D}_{\mathcal{C}}^{\text{train}} \\ \|\boldsymbol{\phi}_i\|^2 & \text{if } \mathbf{x}_i \in \mathcal{D}_{\mathcal{N}}^{\text{train}} \end{cases}, \tag{4.5}$$

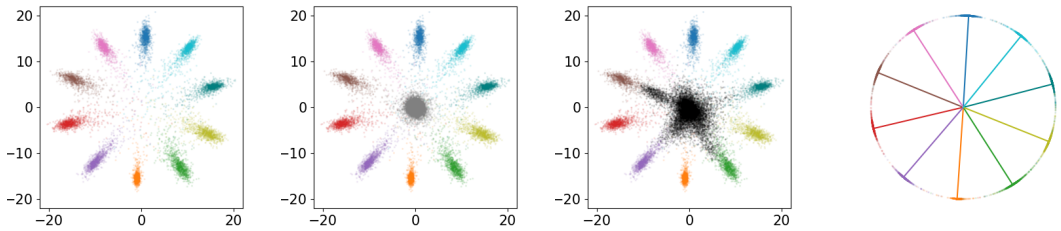
where λ is the weight of the regularization term and s is the target deep feature magnitude.

One challenge that arises regarding this joint loss is how the cross-entropy loss term is supposed to handle negative samples since they do not have a corresponding class label. There are two main ways to address this are: ignoring negative samples for the computation of the cross-entropy loss term or use the EOS loss and adapt it to use a margin. Since the adaption of the EOS loss to employ a margin is itself a new loss (see Section 4.3) and since we observe it to behave slightly differently from the goal of the OS-regularizer — *i. e.* not only do negatives not get drawn to the origin but actually get cast to large magnitudes — we opt for the former. This is also in the interest of being able to examine the effect of the addition of the symmetric OS-regularizer on the SFN-Margin losses (4.3) in isolation.

One benefit the Margin-OS losses bring is the separation of responsibilities: the symmetric OS-regularizer's sole responsibility is discriminating between known and unknown samples, while the cross-entropy loss term's sole responsibility is correctly classifying the known samples (albeit not entirely independently since its classifications depend on the soft feature normalization performed by the symmetric OS-regularizer). While we did not further explore this, in theory we hypothesize that the weight λ now can be used to weigh the importance of the two goals of OSC, this is however subject for future work.

To analyze the effect of the cosine and angular margins in conjunction with the symmetric OS-regularizer individually, we also consider the same loss as (4.4) and (4.5) but with the margin set to zero. This results in a unique loss as the only difference between Cos-OS and Arc-OS is the placement of the margin. We call this loss *Norm-OS* which is simply a softmax loss with symmetric OS-regularization and normalized logit weights, $\|\mathbf{W}_c\| = 1,\ \forall c \in \mathcal{C}$.

Figure 4.2 illustrates the behavior of the deep features of Arc-OS on our toy protocol As is to be expected, it shows in achieving better separation between known and negative/unknown samples compared to SFN-ArcFace (Figure 4.1) while preserving the margin between the classes (Figure 4.2d).



(a) Known (b) Negatives (c) Unknowns (d) Normalized Features

Figure 4.2: DEEP FEATURE DISTRIBUTIONS FOR THE ARC-OS LOSS. *Arc-OS deep feature distributions of the 10 known classes are shown in (a) as learned by the Arc-OS loss . In, (b) and (c) the deep features for negative samples (gray) and unknown samples (black) are superimposed, respectively. Finally, (d) shows normalized features, i. e., being projected onto a hypersphere, with the respective class centers* $\mathbf{W}_c$ *(lines), which are normalized to fit the scale of the features for illustrative purposes.*

4.3 Margin-EOS Losses

Our second proposed generic loss to address research question RQ2 is the *Margin-EOS* loss. Margin-EOS makes use of the CosFace (3.15) and ArcFace logits (3.16) — notably with HFN — and incorporates the EOS loss instead of the softmax loss to include negative samples for training. We choose HFN over SFN as experiments on the toy protocol suggest that it provides superior performance (see Section 6.1). The specific Margin-EOS losses for *Cos-EOS* and *Arc-EOS* are given by:

$$\mathcal{J}_{\text{Cos-EOS}} = \mathcal{J}_{\text{EOS}}(\sigma(L_{\text{CosFace}}(\boldsymbol{\phi}_i))) \text{ and} \tag{4.6}$$

$$\mathcal{J}_{\text{Arc-EOS}} = \mathcal{J}_{\text{EOS}}(\sigma(L_{\text{ArcFace}}(\boldsymbol{\phi}_i))), \tag{4.7}$$

where $\boldsymbol{\phi}_i = B(\mathbf{x}_i)$. Notably, the CosFace and ArcFace logits can naturally be applied to negative samples as they simply never will add a margin because $\delta_{c\mathbf{y}_i} = 0$ for all negative samples $\mathbf{x}_i$.

What makes the Margin-EOS approaches unique is that they are not only forced to discriminate between known classes only via the angle, but also between known and unknown/negative samples.

The EOS loss encourages negative samples to have a uniform distribution in the softmax scores which requires the logit values for all classes to be identical (Dhamija et al., 2018). The only way in which the Margin-EOS losses can achieve a uniform distribution, is by learning an implicit class center for negative samples, *i. e.* learning a deep feature vector that has the same angle to all class centers of known classes, *i. e.*, for $\mathbf{x}_i \in \mathcal{D}_{\mathcal{N}}^{\text{train}}$ we have $\theta_{i,c} = \theta, \forall c \in \mathcal{C}$ for some constant $\theta \in [0, \pi]$. This makes the Margin-EOS approach akin to a background class approach as it learns a non-zero vector as implicit class center for the negatives, unlike EOS, OS, and Margin-OS which learn the zero-vector. However, evaluation is still performed by thresholding the softmax scores and as such it is not a background class approach.

Similar to the previous approaches we introduce a zero-margin version of the Margin-EOS approach that serves as baseline to evaluate the effect of the respective margins in isolation by setting the margin the margin to zero. Following our naming scheme, we call this loss the *Norm-EOS*.

Chapter 5

Experimental Setup

In this chapter we introduce the general experimental setup which concerns both, the preliminary as well as the main experiments. We start by discussing the protocols used for both experiments. Then we introduce the OSCR as our evaluation metric. Finally, we discuss the neural network architectures used as well as our hyperparameter choices for all experiments.

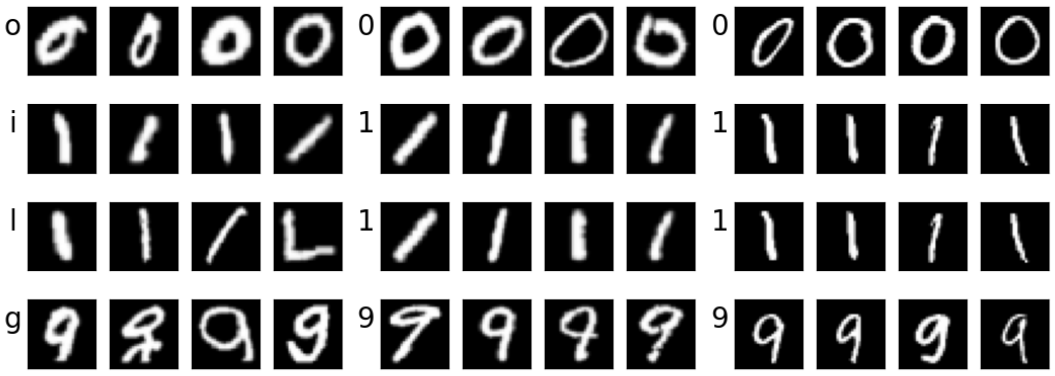
5.1 Protocols

5.1.1 Toy Protocol

The toy protocol is inspired by similar protocols and datasets used for visualization purposes by Wen et al. (2016), Liu et al. (2017), Wang et al. (2018b), Zheng et al. (2018), Dhamija et al. (2018), and Deng et al. (2019). While some use a subset of identities of face recognition datasets, many use the MNIST digits as the known classes. Dhamija et al. (2018) additionally use EMNIST letters as negative classes and Devanagari letters (Acharya and Gyawali, 2016) as unknowns for their OSC toy protocol.

MNIST vs. EMNIST MNIST Instead of mixing MNIST dataset (LeCun et al., 1998) with the EMNIST Letters dataset (Cohen et al., 2017), we opt for using EMNIST MNIST as known classes and EMNIST Letters for the negatives and unknowns to make sure that all samples are converted identically from the original NIST data. Although Cohen et al. (2017) follow the conversion steps outlined by LeCun et al. (1998) to replicate the MNIST conversion process for all EMNIST datasets, they acknowledge some differences in the conversion process (*e. g.* different downsampling methods). Figure 5.1 demonstrates our findings that, based on visual inspection of various samples, the EMNIST Letters (Figure 5.1a) and EMNIST MNIST (Figure 5.1b) seem visibly more blurry compared to MNIST (Figure 5.1c). As we want to avoid making the already simple toy example not artificially simple by allowing networks to pick up on blurryness as indicator if a sample is known or unknown we choose EMNIST MNIST over MNIST, although we acknowledge that EMNIST MNIST apparently constitutes a "more separable problem than the original MNIST dataset." (Cohen et al., 2017)

Removing Visibly Indistinguishable Letters The MNIST Letters dataset consists of 26 classes of letters "a" through "z" and each class contains a mix of uppercase and lowercase letters (Cohen et al., 2017). A visual inspection of individual samples reveals that certain letters (Figure 5.1a) are visually indistinguishable from certain digits (Figure 5.1b and Figure 5.1c). For example, many images depicting the letter "o" are indistinguishable from the digit "0". Similar observations can



(a) EMNIST Letters (b) EMNIST MNIST (c) MNIST

Figure 5.1: COMPARISON EMNIST AND MNIST DATA. *Comparison of random samples of letters ("o", "i", "l", and "g") and digits ("0", "1", and "9") from the EMNIST and MNIST datasets. We show the EMNIST Letters (Figure 5.1a), EMNIST MNIST (Figure 5.1b), and MNIST (Figure 5.1c). EMNIST samples seem visibly more blurry than MNIST samples. The selected letters and digits are arranged per row to highlight which letters are often indistinguishable from the corresponding digits. Rows 2 and 3 depict the same samples for EMNIST MNIST and MNIST.*

be made for the letters "i" and "l" being indistinguishable from digit "1" and letter "g" and digit "9" respectively. This introduces undesirable artifacts that render the dataset impractical for our use case, *e. g.*, this leads to the letters "i" and "l" dominating the training data and every method learning to reject every sample from class "1". This is a problem of the separability of the knowns and negatives and not about the performance of the algorithms, which is why we remove these letters from the data and split the letters into two sets: Negatives (first 11 letters, excl. "o", "i", "l") and unknowns (last 11 letters, excl. "g"). While excluding the letter "g" from the unknowns is arguably an arbitrary choice, it simplifies the visual inspection of the deep features by reducing the overlap of the two classes which is a known effect.

Excluding these letters can be viewed as an arbitrary choice, but we belief that this is in the interest of providing cleaner and easier to interpret visualizations. Also, since the preliminary experiments aim to compare the various methods, we are not interested in the absolute performances, and we belief that these changes do not significantly affect the relative performances.

Breakdown of the Protocol The protocol comprises of 140'400 samples in total and is split into three partitions: training data, validation data, and test data. Table 5.1 provides an overview over the composition of the individual partitions in terms of known, negative, and unknown samples. Since the unknown samples simulate the samples for which no labeled data is available, they occur exclusively in the test set. The training data is composed of 53% known samples and The negative samples make up 47% of the training data (excl. validation). Correspondingly, in the ImageNet open-set protocols the negatives make up 37% (P_1), 52% (P_2), and 39% (P_3) of the respective training data.

Table 5.1: BREAKDOWN OF THE TOY DATASET. *The dataset contains 140400 data points and is divided into three partitions: training set (64.27%), validation set (16.07%), and test set (19.66%). In parenthesis we provide the percentage of the sample type (column; known, negative, and unknown) for each partition (row), e.g., the known samples make up roughly 53% of the total training set.*

	Known	Negative	Unknown	Total
Training	48'000 (53.19%)	42'240 (46.81%)	0 (0%)	90240
Validation	12'000 (53.19%)	10'560 (46.81%)	0 (0%)	22560
Test	10'000 (36.23%)	8'800 (31.88%)	8'800 (31.88%)	27600

5.1.2 ImageNet Open-Set Protocols

Palechor et al. (2023) propose — based on the Master's thesis by Bhoumik (2021) — three protocols with varying degrees of difficulty for evaluating open-set classification algorithms. These protocols consist of subsets of the ImageNet classes (Russakovsky et al., 2015) and are each grouped into known, negative, and unknown classes. The three protocols P_1, P_2, and P_3 are designed to have increasing levels of difficulty by having increasing levels of similarity in appearance and overlap in visual features between inputs from known and unknown classes (Palechor et al., 2023).

Protocol P_1 consists of $C = 116$ known classes all of which are various classes of dogs. The negative classes consist of 67 classes of other 4-legged animal classes. The unknown classes consist of 166 non-animal classes. As such, P_1 poses an easy task for discriminating knowns and unknowns, because they are semantically very different and share little visual features. As such, it is well suited to test the evaluation of out-of-distribution detection algorithms. However, P_3 poses a hard task for closed-set classification.

Protocol P_2 is the smallest of the three protocols in terms of data points with only $C = 30$ known classes, depicting half of the hunting dog classes. The negatives are made up of the second half of the hunting dog classes, *i.e.* it contains 31 classes. The unknowns consist of 55 classes of other 4-legged animals. Being the smallest in size allows this network to be used for optimizing hyperparameters which can be transferred to protocols P_1 and P_3 (Palechor et al., 2023). Since this is a comparative study, we did not do this. P_2 sits between P_1 and P_3 in terms of difficulty of open-set and closed-set classification difficulty.

Lastly, protocol P_3 consists of $C = 151$ known classes, 97 negative classes, and 164 unknown classes. All of these classes contain a mix of various classes such as animals, plants, and other objects, making this the hardest task for open-set classification but the simplest task for closed-set classification. Because of the similarities of the known and unknown classes "it is very unlikely that out-of-distribution detection algorithms are able to discriminate between them, and real open-set classification methods need to be applied." (Palechor et al., 2023)

5.2 Evaluation Metric

We evaluate the performance of all losses on the open-set protocols via the OSCR curves introduced by Palechor et al. (2023) and corrected by Bisgin et al. (2023). The OSCR curves are a combination of two metrics which handle known and mixed unknown (*i.e.* negative and unknown) samples separately: the Correct Classification Rate (CCR) and the False Positive Rate (FPR), both of which are defined as functions of the probability threshold τ. Following Bisgin et al. (2023)

CCR and FPR are defined as:

$$\text{CCR}(\tau) = \frac{|\{\mathbf{x}_i | \mathbf{y}_i \leq C \wedge \arg\max_{1 \leq c \leq C} \mathbf{p}_{i,c} = \mathbf{y}_i \wedge \mathbf{p}_{i,c} \geq \tau\}|}{|\mathcal{D}_{\mathcal{C}}^{\text{test}}|} \in [0, 1] \tag{5.1}$$

$$\text{FPR}(\tau) = \frac{|\{\mathbf{x}_i | \mathbf{y}_i > C \wedge \max_{1 \leq c \leq C} \mathbf{p}_{i,c} \geq \tau\}|}{|\mathcal{D}_{\mathcal{M}}^{\text{test}}|} \in [0, 1], \tag{5.2}$$

where $\mathbf{y}_i \leq C$ indicates a known sample, *i. e.*, $\mathbf{y}_i \in \mathcal{C}$, and $\mathbf{y}_i > C$ indicates a negative or unknown sample, *i. e.*, $\mathbf{y}_i \in \mathcal{M}$. The CCR is very similar to the accuracy and extends its idea to an open-set problem. It measures the proportion of all known test samples that are classified as known ($\mathbf{p}_{i,c} \geq \tau$) and the sample is correctly classified ($\arg\max_{1 \leq c \leq C} \mathbf{p}_{i,c} = \mathbf{y}_i$). The FPR measures the proportion of all negative and unknown test samples that are classified as known. The OSCR curves have the advantage over other metrics that the CCR and FPR measure the two goals of open-set recognition directly.

The OSCR are drawn by plotting the CCR against the FPR and increasing the threshold from 0 to 1, which draws the curves from right to left, *i. e.*, from high to low FPR values. As such, they depict CCR values for a given FPR value (CCR@FPR). We plot the FPR on a logarithmic curve to highlight low FPR values because most applications require a very few false positives Bisgin et al. (2023).

Interpretation Because the OSCR curve combines two metrics and plots them against each other while being a function of a threshold that is not explicitly depicted in the OSCR curve it can be hard to interpret. Crucially, we always want the CCR to be as close to 1 as possible, while we want the FPR to drop from 1 to 0 as fast as possible. As such, the perfect open-set classifier will draw a curve from the top right (CCR $= 1$ at FPR $= 1$) to the top left (CCR $= 1$ at FPR $= 10^{-4}$). For example, consider the toy experiment in Figure 6.3a, where the Cos-EOS classifier achieves almost perfect classification on known samples and perfect separation of known and negative samples. Note that in all figures we plot the FPR from 10^{-4} to 1 but report the exact CCR@FPR values in Table 6.2 only from 10^{-3} as no classifier reached 10^{-4} exactly.

An important special case to consider is the case when we set $\tau = 0$, for which we have that $\text{FPR}(0) = 1$, *i. e.* the classifier classifies every single sample as known, and $\text{CCR}(0) = \text{Acc}$, where Acc is the closed-set accuracy, *i. e.* the accuracy of the closed-set classifier when evaluated only on the known samples. This lets us analyze the research sub-questions for RQ1 and RQ2 respectively within the same plot and using the same evaluation metric. Therefore, we can interpret the CCR@FPR values as the closed-set accuracy for FPR $= 1$, which is always at the rightmost edge of any OSCR curve. This is also the point from which the curves start being drawn when increasing the threshold from 0 to 1 and the curves then extend to the left. For approaches where the maximum softmax score of some sample and for any class reaches 1 exactly — up to any reasonable precision — there is no threshold τ that allows to threshold these scores, and thus FPR values cannot be lowered any further (Bisgin et al., 2023). In this case the OSCR curve will not extend further to the left and, for example, highlight issues when s is too large (see Section 3.2.2).

5.3 Neural Networks

Since this is a comparative study of loss functions we keep the network backbone architecture identical across all experiments on the ImageNet protocols and toy protocol, respectively. For the toy experiments we use the LeNet++ backbone following Wen et al. (2016) and Dhamija et al. (2018). For the ImageNet experiments we use a ResNet-50 backbone He et al. (2016) as this is a

Table 5.2: HYPERPARAMETERS OVERVIEW. *This table shows the hyperparameters used for the ImageNet protocols P_1, P_2, and P_3 and the toy protocol P_{toy}. C denotes the number of known classes, K denotes the deep feature dimensionality, and s denotes the scaling factor or target for the deep feature magnitude. The margins are denoted by m_{mult} (multiplicative angular margin), m_{cos} (additive cosine margin), m_{ang} (additive angular margin), and m_{logit} (additive logit margin). λ denotes the weight for the regularization terms.*

Protocols	K	C	s	m_{mult}	m_{cos}	m_{ang}	m_{logit}	λ
P_1	116	116	17		0.35	0.5		0.01
P_2	30	30	16		0.35	0.5		0.01
P_3	151	151	18		0.35	0.5		0.01
P_{toy}	10	10	13	1.2	0.35	0.5	0.3	0.01
P_{toy} (vis)	2	10	13		0.15	0.4		0.01

common network architecture for image classification tasks and also used by Palechor et al. (2023) and Bisgin et al. (2023) on which our experiments build. We train all networks from scratch and do not make use of pre-trained models as these include knowledge of ImageNet samples and classes that we use as unknown samples at test time. For both network topologies we discard the head and use the respective heads outlined in Chapter 4.

For all experiments we set the deep feature dimensionality — *i. e.* length of the deep feature vectors — equal to the number of classes, *i. e.* $K = C$ (see Table 5.2). The only exception being the toy experiments used for visualizing the deep features, where we set $K = 2$. Additionally, we do not add a bias to the deep features.

ImageNet Preprocessing Following Palechor et al. (2023) and Bisgin et al. (2023) we perform the following image preprocessing steps on the ImageNet data. We resize the images, perform random crops, and apply a horizontal flip with a flip probability of 0.5.

Implementation Details Our code is publicly available[1] and is a fork[2] of the the code from Palechor et al. (2023) and Bisgin et al. (2023). The implementations for ShpereFace, CosFace, and ArcFace taken from the code[3] by Liu et al. (2023). The implementations of all of our logit functions include the characteristic gradient detachment trick, which, according to Liu et al. (2023), increase the performance of margin-based loss functions. All experiments are run on Nvidia GeForce RTX 2080 Ti GPUs.

5.4 Hyperparameters

An overview over all hyperparameters for each experiment is given in Table 5.2. It lists the scaling factor or target for the deep feature magnitude s, the number of classes C (not a hyperparameter but relevant for s), as well as the multiplicative angular margin m_{mult}, the additive cosine margin m_{cos}, the additive angular margin m_{ang}, and additive logit margin m_{logit}. The logit margin was used only for SM-Softmax in the preliminary experiments, which is why the other cells are left empty.

[1] https://github.com/TwoDigitsOneNumber/openset-imagenet-comparison
[2] https://github.com/AIML-IfI/openset-imagenet-comparison
[3] https://github.com/ydwen/opensphere

Margin The margin m is perhaps the most important hyperparameter for any margin-based loss, yet there is currently no comprehensive understanding or guidance for how to choose them optimally. The margins for SM-Softmax (Liang et al., 2017), SphereFace (Liu et al., 2017), CosFace (Wang et al., 2018b), and ArcFace (Deng et al., 2019) were chosen empirically with some providing lower and upper bounds and results of experiments on the margin parameter. Zhang et al. (2019) conducted a hyperparameter study on m_{cos} for CosFace and m_{ang} for ArcFace, but proceeded to choose $m_{\text{cos}} = 0.25$ and $m_{\text{ang}} = 0.5$ unfortunately without any reasoning. Since optimizing the margin is out of scope for this thesis we choose the hyperparameters proposed in their original publications, *i. e.* $m_{\text{cos}} = 0.35$ and $m_{\text{ang}} = 0.5$ (Wang et al., 2018b; Deng et al., 2019). This comes with the caveat that one might be closer to optimal than the other for the respective methods, requiring cautious interpretation. The logit margin for SM-Softmax for preliminary experiments is set to $m_{\text{logit}} = 0.3$ following the choice of Liang et al. (2017) for their experiments on CIFAR-10. These margins provide reasonable results throughout all experiments. The only exception being Arc-OS which diverged on protocol P_1, however, we do not know if this was caused by the margin.

For the deep feature visualizations we require the margins to satisfy $m_{\text{cos}} \leq 1 - \cos(\frac{2\pi}{C}) \approx 0.1910$ (Wang et al., 2018b) and $m_{\text{ang}} \leq \frac{2\pi}{C} \approx 0.6283$ for $C = 10$. We do not rigorously optimize the margins since they only serve the purpose of visualization but manually choose the margins as large as possible to maximize the visible effect but small enough such that all methods converge. We choose $m_{\text{cos}} = 0.15$ and $m_{\text{ang}} = 0.4$.

Feature Magnitude As discussed in Section 3.2.2 we can compute a lower bound on the feature magnitude s (3.11) as a function of the number of classes C and the expected minimum posterior probability of a class center $\hat{\mathbf{p}}_{i,c}$, the latter being a hyperparameter that can be freely chosen. We choose $\hat{\mathbf{p}}_{i,c} = 0.999995$ as this value allows a perfectly classified sample (*i. e.* $\theta_{i,\mathbf{y}_i} = 0$) to achieve a softmax score of exactly 1 when taking into account the precision of a PyTorch `float32` datatype. We computed s for each protocol as the ceil of the lower bound, giving us the values s as summarized in Table 5.2. Note that for the OS loss we set $\xi = s$ for a fair comparison to all other methods.

Deep Feature Dimensionality For all losses and all protocols the feature dimensionality K is set to the number of known classes, *i. e.* $K = C$, except the toy protocol when used for visualizing the deep features, where we set $K = 2$.

Training Parameters Following Dhamija et al. (2018) and Bisgin et al. (2023) we train all networks on the ImageNet open-set protocols for 120 epochs with a constant learning rate of 10^{-3}. In contrast to Bisgin et al. (2023) we use the SGD optimizer instead of Adam. The SGD optimizer uses a momentum factor of 0.9. The networks on the toy protocols are trained for 15 epochs.

Regularizer Weight Zheng et al. (2018) find $\lambda = 0.01$ to work best for the Ring loss but state that performance does not seem to vary much for larger or smaller values. Notably, their definition of λ for the Ring loss differs by a factor of $\frac{1}{2}$ from our definition, and as such corresponds to a value of 0.005 for us. We find however, that $\lambda = 0.01$ works best for our losses, with 0.1 and 0.001 being too large and too small, respectively, when training the OS loss on protocol P_2. As such we choose $\lambda = 0.01$.

Chapter 6

Experiments

We introduce the preliminary toy experiments and the main experiments on the ImageNet open-set protocols along side the results. Even though all experiments on the ImageNet protocols follow the same procedure and evaluation methods, we discuss them separately based on which research question (RQ1 or RQ2) they address.

6.1 Preliminary Toy Experiments

The toy experiments serve mainly two purposes: (1) they are used for developing our losses and (2) they are used for visualizing deep features to build intuition about the clustering of the deep features for the respective loss functions. Additionally, we have also used the toy protocol for verifying correctness of implementations of our generalized loss functions (*e. g.* `CosineMargin`) against their respective special cases (*e. g.* `CosFace`) from the implementations[1] by Liu et al. (2023).

Using the toy protocol for developing our losses requires it to be similar to the ImageNet open-set protocols proposed by Palechor et al. (2023) in terms of their composition, such as ratios of negatives to the total training data. While the toy protocol poses a task of much lower difficulty, we find that the results are qualitatively similar to findings on P_2 on which we verified some early results. This gives us reason to believe that the toy protocol provides results that are relevant to the ImageNet protocols and thus using the toy protocol for development of the proposed SFN-Margin losses, Margin-OS losses, and Margin-EOS losses is justified, even if this is not guaranteed.

The preliminary experiments provide empirical guidance in cases where theoretical knowledge is insufficient to choose one method over another, *e. g.*, which margin-based loss works best under classification via softmax-scores or whether to penalize feature margins symmetrically or asymmetrically. Furthermore, they provide a way to validate our intuition about certain methods and uncover previously overlooked ideas by drastically reducing the time required to train, analyze, and compare small variations in the methods as well visualize the distributions of the deep features.

We conduct experiments on: (1) what margin to apply, (2) using SFN or HFN, and (3) using the original OS-regularizer or a symmetric OS-regularizer. For the latter two questions only the additive cosine margin and the additive angular margin are considered. All results are discussed in detail in the following sections and summarized in Table 6.1, which depicts the CCR@FPR on the unknowns, *i. e.* $\mathcal{D}_{\mathcal{U}}^{\text{test}}$, where the results are grouped by the question they address. To facilitate the comparison some losses are listed twice and grouped by their margin such that for each column the best performance per group is underlined.

[1] https://github.com/ydwen/opensphere

Table 6.1: TOY PROTOCOL RESULTS. *This table depicts the CCR@FPR for all losses as well as the closed-set accuracy (Acc) on the unknowns. Losses are grouped by the respective experiments.For each column, the best performance per group is underlined. Empty cells indicate that the respective FPR was not reached.*

Group	Loss	CCR@FPR			Acc
		10^{-3}	10^{-2}	10^{-1}	1
Margin Types	SM-Softmax		0.2789	0.8310	0.9921
	SphereFace	0.0302	0.1644	0.8607	0.9927
	CosFace	0.0361	0.2789	0.8900	0.9934
	ArcFace	0.0107	0.1745	0.8801	0.9942
HFN vs. SFN (CosFace)	CosFace	0.0361	0.2789	0.8900	0.9934
	SFN-CosFace	0.0322	0.4377	0.9030	0.9947
HFN vs. SFN (ArcFace)	ArcFace	0.0107	0.1745	0.8801	0.9942
	SFN-ArcFace	0.0617	0.5522	0.9140	0.9939
HFN vs. SFN (Cos-EOS)	Cos-EOS	0.9849	0.9919	0.9923	0.9923
	Cos-EOS (SFN)	0.9579	0.9848	0.9928	0.9930
HFN vs. SFN (Arc-EOS)	Arc-EOS	0.9853	0.9933	0.9933	0.9933
	Arc-EOS (SFN)	0.9677	0.9857	0.9931	0.9933
OS vs. sym. OS (Cos-OS)	Cos-OS	0.8841	0.9503	0.9845	0.9939
	Cos-OS (non-sym.)	0.8548	0.9498	0.9842	0.9933
OS vs. sym. OS (Arc-OS)	Arc-OS	0.8779	0.9483	0.9851	0.9936
	Arc-OS (non-sym.)	0.8769	0.9499	0.9857	0.9933

6.1.1 Comparing Margin Types

Testing all possible combinations of margin-based losses with losses that include negative samples, even on the smallest ImageNet protocol P_2, is infeasible due to the combinatorial explosion in the number of possible combinations. Hence we aim to consider only the most promising margin-based losses — or margin types. We compare SM-Softmax (additive logit margin), SphereFace (multiplicative angular margin), CosFace (additive cosine margin), and ArcFace (additive angular margin).

Figure 6.1 shows the OSCR curves for all losses evaluated on the negatives and unknowns of the test set respectively. While all losses reach FPR values of almost 10^{-4} on the negatives and unknowns, SM-Softmax fails to do so, most likely due to the fact that this is the only method which does not employ feature or weight normalization. SM-Softmax also consistently underperforms in terms of CCR@FPR compared to all other margin types on the unknowns, except for FPR $= 10^{-2}$ where it achieves the best performance together with CosFace (see Table 6.1). On the negatives, both SM-Softmax and SphereFace, consistently are outperformed by CosFace and ArcFace. Similarly, SphereFace is outperformed on the unknowns. CosFace achieves the best performance across all non-zero thresholds. ArcFace which achieves the highest closed-set performance with an accuracy of 0.9942%, 0.0008% higher than CosFace.

In summary, the additive cosine margin and additive angular margin of CosFace and ArcFace seem better suited for open-set classification tasks and are thus our two margins of choice. Due to time constraints we did not further explore the modifications of SM-Softmax as this would have required more understanding of the behavior of the method, which is beyond the scope of this thesis.

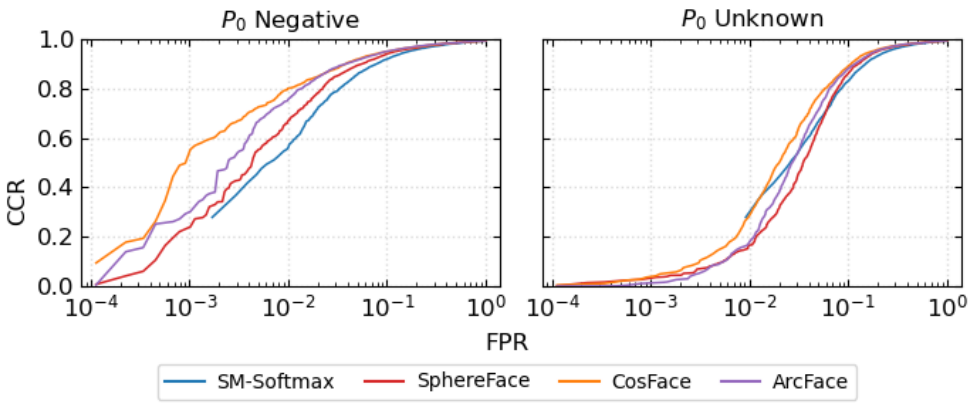


Figure 6.1: OSCR Curves for Preliminary Experiments: Margin Types. *This figure shows the OSCR curves of the preliminary experiments comparing different margin-based losses with different margin types: SM-Softmax, SphereFace, CosFace, and ArcFace. The curves depict the performance evaluated on the negative and unknown test samples.*

6.1.2 Hard vs. Soft Feature Normalization

We compare hard and soft feature normalization for the margin-based losses CosFace and ArcFace (Section 4.1) and the Margin-EOS losses (Section 4.3).

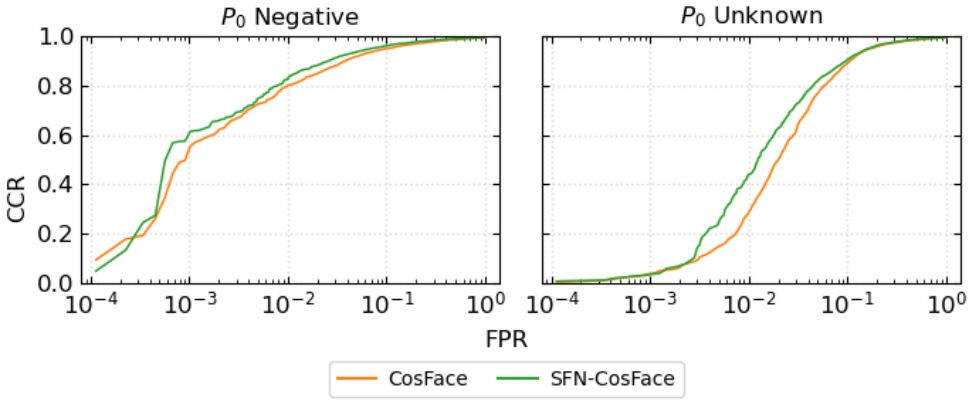
SFN-Margin Losses Figure 6.2 shows the OSCR curves on the negatives and unknowns from the test set for HFN and SFN with the CosFace and ArcFace loss, respectively. The closed-set accuracy is very comparable between all approaches and differences in CCR@FPR values become generally larger for FPR values of 10^{-1} and 10^{-2}. These differences between HFN and SFN are smaller for the cosine margin, whereas the angular margin benefits strongly from SFN.

While the difference in closed-set accuracy are very small and potentially insignificant, the general trend and differences at lower FPR values show a clear trend that SFN should be preferred over HFN for these margin-based losses. This confirms on a small-scale experiment the findings by Zheng et al. (2018) and Liu et al. (2023) under a different evaluation method.

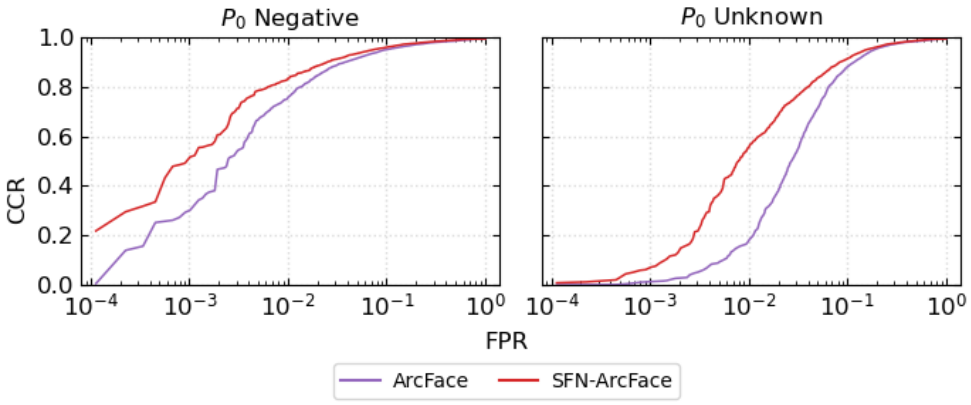
Margin-EOS Losses Figure 6.3 shows the OSCR curves for the Margin-EOS losses on the negative and unknown test samples with HFN and SFN, respectively. The differences are almost not visible as all losses manage to almost perfectly separate knowns from negatives/unknowns while correctly classifying all known classes. However, for both margin types the performance of the HFN versions surpasses the Margin-EOS version that employs SFN. For this reason we choose HFN, which makes our Margin-EOS losses, in contrast to all other losses considered in this thesis, the only one that cannot support the classification via deviations in the feature magnitudes.

6.1.3 OS-Regularizer vs. Symmetric OS-Regularizer

Figure 6.4 shows the Margin-OS losses with symmetric OS-regularizer, as introduced in Section 4.2, and with the original — *i.e.* non-symmetric — OS-regularizer. Both regularizations achieve almost identical CCR@FPR values for all levels FPR values and the differences are likely not significant. The only exception being the CCR at FPR $= 10^{-4}$ where the CCR of the symmetrically penalized version outperforms the non-symmetric by about 20%. We hypothesize that this



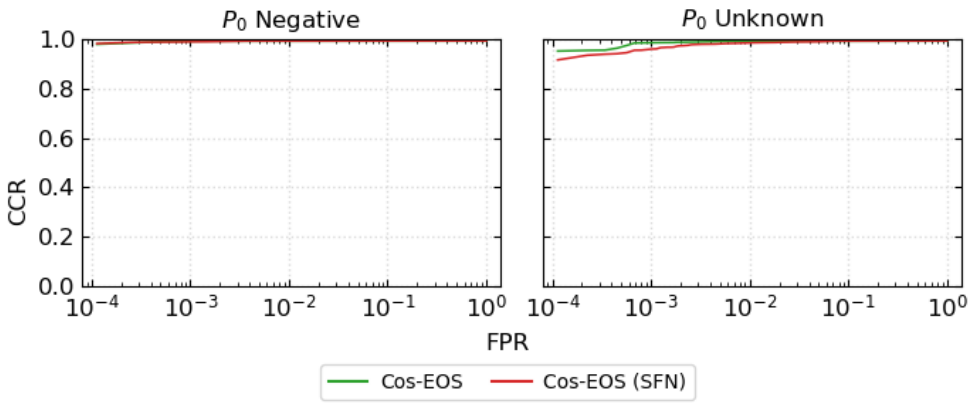
(a) HFN versus SFN for the CosFace



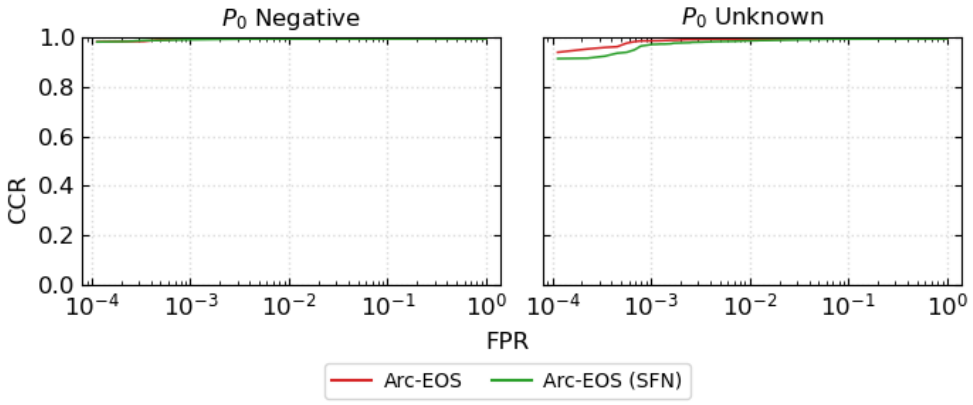
(b) HFN versus SFN for the ArcFace

Figure 6.2: OSCR CURVES FOR PRELIMINARY EXPERIMENTS: HFN VS. SFN. *This figure shows the OSCR curves of the preliminary experiments comparing HFN and SFN for the additive cosine margin in Figure 6.2a and additive angular margin Figure 6.2b. The curves depict the performance evaluated on the negative and unknown test samples.*

outperformance for thresholds of 1 might be due to the fact that the non-symmetric penalty yields more unknown samples to reach a large enough feature magnitude such that its maximum softmax score equals 1 and thus cannot be discriminated from the known samples with potentially larger magnitudes. Recalling the discussion in Section 3.2.2, both penalties encourage the magnitudes to be as large as necessary, but only the symmetric penalty encourages the magnitudes to be as small as possible. This observation leads us to prioritize the symmetric penalty because it brings potential upsides while, to the best of our knowledge, not providing any downsides. Additionally, the symmetric penalization lends itself more naturally to the idea of feature normalization as it defines a clear target magnitude as opposed to an unbound range of acceptable magnitudes.



(a) HFN versus SFN for the Cos-EOS



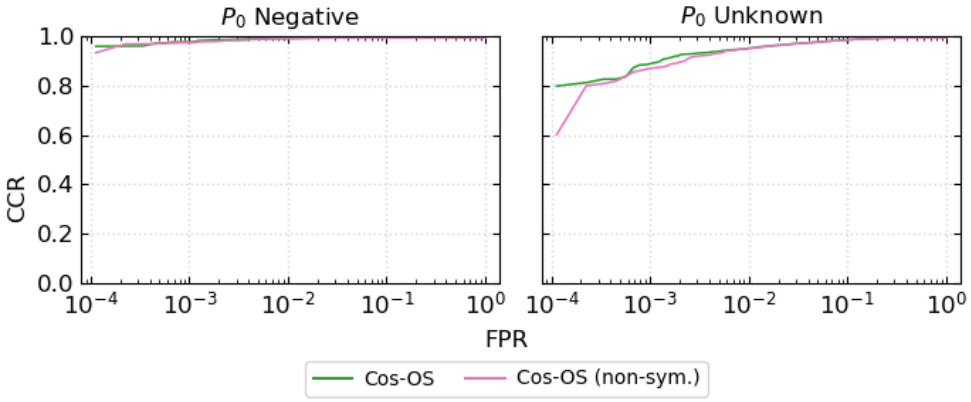
(b) HFN versus SFN for the Arc-EOS

Figure 6.3: OSCR CURVES FOR PRELIMINARY EXPERIMENTS: MARGIN-EOS FEATURE NORMALIZATION. *This figure shows the OSCR curves of the preliminary experiments comparing HFN and SFN for the Margin-EOS losses: Cos-EOS Figure 6.3a and Arc-EOS Figure 6.3b. The curves depict the performance evaluated on the negative and unknown test samples.*

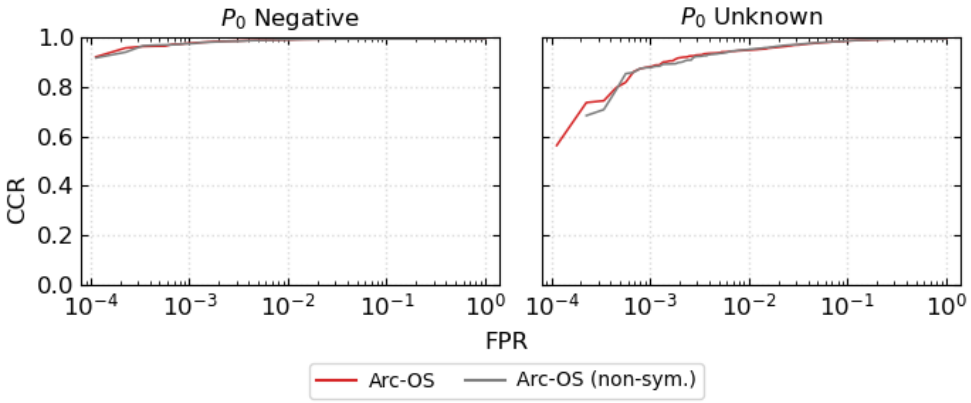
6.1.4 Deep Feature Visualizations

The experiments for computing visualizations of the deep feature distributions are identical to the toy experiments except that we set the deep feature layer to become a bottleneck in the network. In other words, we set the deep feature dimensionality — *i. e.* the length of the deep feature representation vector — to $K = 2$ such that we can plot thee deep features $\phi_{i,1}$ against $\phi_{i,2}$.

Crucially, we cannot visualize the deep features of the Margin-EOS methods as these learn a deep feature vector for the negative samples that has equal angles to all known class centers. However, for $C = 10$ this is not possible in two dimensions.



(a) Cos-OS versus Non-symmetric Cos-OS



(b) Arc-OS versus Non-symmetric Arc-OS

Figure 6.4: OSCR CURVES OF PRELIMINARY EXPERIMENTS: OS-REGULARIZATION. *This figure shows the OSCR curves of the preliminary experiments comparing the symmetric OS-regularizer (Cos-OS in Figure 6.4a and Arc-OS in Figure 6.4b) against the non-symmetric counterparts for the Margin-OS approaches. The curves depict the performance evaluated on the negative and unknown test samples.*

6.2 ImageNet Experiments

The ImageNet experiments are the main experiments conducted in this thesis and are used to answer our research questions. We analyze the effect of the additive cosine margin and the additive angular margin on losses trained and evaluated on the ImageNet open-set protocols. We compare the performance of our proposed loss functions to the respective benchmarks and to respective zero-margin versions of our losses to analyze the effect of the margin in isolation to the normalization of weight and feature magnitudes. The zero-margin versions are required because the normalizations are fundamentally required for all of our losses and make a direct comparison with softmax, EOS, and OS difficult.

Since all experiments are identical from a practical perspective and only differ in the choice of loss functions, we group our losses as: SFN-Margin, Margin-OS, and Margin-EOS with an additional group for the benchmarks (see Table 6.2). The SFN-Margin losses address research

question RQ1 and are compared to the softmax loss, which serves as our benchmark, since neither of these methods incorporate negative samples during training. The two groups Margin-OS and Margin-EOS address research question RQ2 and are compared to the EOS and OS loss. To keep the OSCR plots uncluttered we omit the EOS and only compare the results visually against the OS since both achieve very comparable performance with the OS loss usually slightly better.

Table 6.2 shows the CCF@FPR values on unknown test samples for each protocol. Each table shows all loss functions in a single table in order to facilitate the comparison of all methods beyond the individual research questions and to present all results compactly.

Angle Distributions All margin-based and respective zero-margin losses are forced to discriminate classes solely based on the angle to the respective class centers, with exception of the effect of soft feature normalization. For this reason we analyze the behavior of known, negative, and unknown samples in terms of the angles in more depth as well. Figure 6.6, Figure 6.8, and Figure 6.10 show the distributions of angles for the SFN-Margin, Margin-OS, and Margin-EOS losses. Each plot depicts the histograms of the angles of known samples to the ground-truth class center, *i. e.* $\theta_{i,\mathbf{y}_i}$. For negative and unknown samples, which do not have a ground-truth class center, the angle to the closest class center (in terms of the angle) is depicted, *i. e.* $\min_{c\in\mathcal{C}}\{\theta_{i,c}\}$.

For losses with feature and weight normalization we can compute the probability curve as s function of the angle between the deep feature and a class center (Zhang et al., 2019). We superimpose these probability curves with consideration of the respective scaling factor s used for each protocol. For margin-based losses we additionally superimpose the probability curve with consideration of the margin that is applied during training (dashed black line). The cosine and angular margins mainly shift the probability curves to the left and as such encourage learning discriminative features (Zhang et al., 2019). The probability curves visually indicate the softmax scores that are achieved at a given angle.

Recalling the assumptions of the probability curves (3.10), we see that the assumption on the HFN is not fulfilled for any SFN-Margin loss, because the feature magnitudes can show large deviations from the target feature magnitude s. However, for most protocols this assumption is at least fulfilled for the average sample. Since this assumption is not fulfilled for each individual sample, we need to analyze (and later interpret) these curves with caution. Ideally we would like to see the known distributions clustered as far to the left as possible and the negative and unknown distributions clustered around $\frac{\pi}{2}$ since this is the empirical upper bound on the angles.

6.2.1 SFN-Margin Losses

Figure 6.5 shows the OSCR curves of the SFN-Margin losses and the softmax loss on all three ImageNet open-set protocols.

Protocol 1 On protocol P_1 we can clearly see that the SFN-CosFace and SFN-ArcFace achieve superior closed-set accuracy over SFN-Norm and softmax. This observation holds on the negative and on the unknown samples and is important, considering that P_1 is the hardest protocol for closed-set classification. SFN-CosFace in fact achieves the highest closed-set accuracy on the negative samples out of all loss functions evaluated on P_1. Training of SFN-ArcFace is a bit more unstable on protocol P_1 compared to the other losses which might have lead to suboptimal performance, however, we find no sign of divergence.

For negative samples and FPR values in the range from roughly 2×10^{-2} to 1, SFN-CosFace and SFN-ArcFace outperform SFN-Norm and softmax, while underperforming at lower FPR values. On the unknowns we observe a similar effect, with the margin-based losses achieving higher CCR for FPR values in the range of roughly 3×10^{-3} to 1. For lower FPR values SFN-CosFace

achieves similar performance to softmax and SFN-Norm while the CCR@FPR values of SFN-ArcFace deteriorate quickly, reaching CCR $= 0$ at roughly 6×10^{-4}. SFN-Norm, which normalizes the weights and uses SFN for feature normalization, achieves almost identical performance as the benchmark for all FPR values, however, the CCR@FPR values are constantly about 0.01 to 0.02 lower. As expected, all losses are able to reject unknowns better than negatives as they are semantically further away from the known classes than the negatives.

Protocol 2 On the intermediate protocol P_2, SFN-CosFace and SFN-ArcFace also achieve higher closed-set accuracy of roughly 0.67 compared to roughly 0.62 for softmax and SFN-Norm on the unknowns. A similar observation can be made on the negatives.

The difference between the performance on the negatives and unknowns is very small and visually indistinguishable, which is to be expected, since they contain semantically similar classes that share similar visual features. The only notable difference being that the OSCR curves of the margin-based losses don't extend beyond FPR $= 2 \times 10^{-3}$ whereas SFN-Norm and softmax reach FPR values of 7×10^{-4}. On negative and unknown samples SFN-CosFace performs worse than SFN-ArcFace. Overall, all losses achieve similar CCR@FPR values for FPR values smaller than roughly 10^{-2} with SFN-CosFace and, in particular, SFN-ArcFace outperforming SFN-Norm and softmax for larger FPR values.

Protocol 3 The most difficult protocol for OSC, P_3, both margin-based approaches achieve almost identical CCR@FPR values with maximal differences of only about 0.02 on negatives and unknowns. Both achieve a high closed-set accuracy of about 0.79, which is amongst the highest closed-set accuracies achieved on P_3 for all losses and only surpassed by Cos-OS. The closed-set accuracy of SFN-Norm and softmax is slightly below 0.76

On the negative and unknown samples all SFN-Margin losses achieve a lower FPR than the benchmark softmax, *i.e.*, their curves extend further to the left. On the negative samples the benchmark fails to extend to FPR values of 10^{-2} while SFN-Norm and the margin-based losses reach values of 5×10^{-3} or lower. On the unknown samples SFN-CosFace and SFN-ArcFace achieve comparable performance to the negative samples. Notably, SFN-Norm achieves almost identical CCR@FPR values to softmax but reaches FPR values of 10^{-3} compared to softmax which stops at about 7×10^{-3}. For for FPR values below 10^{-1}, the CCR@FPR values of the margin-based losses are significantly lower compared to softmax or SFN-Norm.

Angle Distributions Figure 6.6 shows the angle distributions for the SFN-Margin losses. As expected, imposing a margin on the angle forces the networks to draw known samples closer to the respective class centers compared to SFN-Norm and softmax. On all three protocols SFN-CosFace and SFN-ArcFace achieve angles of known classes to the class centers that are smaller than $\frac{\pi}{4}$, with the largest angles being around $\frac{\pi}{2}$. SFN-Norm and softmax generally have angles in the range of $[\frac{\pi}{4}, \frac{\pi}{2}]$ for known, unknown, and negative samples. On all protocols the margin-based approaches shift the peak of the knowns closer to 0 than approaches without a margin. This appears to separate the knowns from the negatives and unknowns better, especially on P_1, although it is not very clear. However, the margin-based losses also shift the distributions of the negatives and unknowns towards the left. This leads to negative and unknown samples reaching similarly small angles and more overlap of the distributions for lower angles. As such the distributions of the angles between knowns and negatives/unknowns, respectively, visually seem more separated for softmax and SFN-Norm on all protocols.

Interestingly, the distribution of angles of the known classes are heavily right-skewed and extending up to angles around $\frac{\pi}{2}$. This is likely an empirical upper bound on the angles as we know that the average angle of a known sample to any non-ground-truth class center is almost exactly $\frac{\pi}{2}$. This indicates large potential for misclassifying samples, since — *ceteris paribus* — many

negative and unknown samples would achieve higher softmax scores than many of the known samples.

Summary On all three protocols SFN-CosFace and SFN-ArcFace achieve higher closed-set accuracies than to SFN-Norm and softmax, which achieve similar accuracies. SFN-CosFace and SFN-ArcFace outperform the others for high FPR values but their OSCR curves tend to drop faster for decreasing FPR values, with exception of protocol P_2. With increasing open-set difficulty from one protocol to the next, the range of FPR values on which SFN-CosFace and SFN-ArcFace outperform the approaches without margins becomes smaller.

The margin-based losses learn smaller angles than losses without margins as they draw the knowns closer to angles of 0. However, compared to softmax and SFN-Norm, SFN-CosFace and SFN-ArcFace have more negative and unknown samples with very low angles to class centers which leads to more overlap of the distributions on the low end of the angles.

6.2.2 Margin-OS and Margin-EOS Losses

The OSCR curves for the Margin-OS losses and Margin-EOS losses are shown in Figure 6.7 and Figure 6.9, respectively. We analyze the results for each protocol separately, providing a summary in the end. Figure 6.8 and Figure 6.10 shows the angle distributions for the Margin-OS and Margin-EOS losses, respectively.

Protocol 1 For protocol P_1 the Arc-OS loss was not able to converge, showing high fluctuations in the validation loss and constant training loss. Unfortunately, we do not know what caused this failure to converge.

Cos-OS and Norm-OS achieve similar closed-set accuracies of about 0.7 which slightly surpass the benchmark accuracies on the unknown and negative samples. The Margin-EOS losses achieve almost identical closed-set accuracy of 0.71 with Norm-EOS being slightly lower. These small differences are not surprising on P_1 given that it poses the hardest closed-set task of all three protocols.

When considering the open-set performance we can see clear differences between the performance on the negatives and unknowns for Cos-OS. On the negative samples the Cos-OS OSCR curve drops below the curves for Norm-OS and OS at around 10^{-1} and shows consistently worse CCR@FPR values for all lower FPR values. However, Cos-OS seems much better at generalizing towards the unknowns as it clearly and consistently achieves higher CCR@FPR for all FPR values above 10^{-2} compared to Norm-OS and OS. But again, it shows a clear drop in performance for FPR values below 10^{-3}. Norm-OS shows very similar performance to OS on the unknowns with slightly worse performance on the negatives.

The OSCR curves for Cos-EOS and Arc-EOS show that they are able to achieve slightly higher CCR@FPR values on the negatives than the benchmark for almost all levels of FPR. This advantage largely disappears on the unknowns for FPR values below about 5×10^{-3} for both losses, after which their performance deteriorates and lags behind the OS benchmark. Norm-EOS again shows qualitatively similar results to the benchmark but shows consistently lower CCR, especially on the unknown test samples.

Protocol 2 Norm-OS achieves the highest closed-set accuracy of 0.68 on the unknowns of protocol P_2 out of all losses, with Cos-OS close behind. While Arc-OS was able to converge on P_2, its loss curves again reveal some training instabilities, leading to having among the worst accuracy of less than 0.63. We can see similar results on the negative samples. Cos-EOS and Arc-EOS

achieve very similar closed-set accuracies of about 0.67, and are thus between Norm-OS and Cos-OS, with Norm-EOS performing noticeably worse but still surpassing both benchmarks EOS and OS.

The training instabilities of Arc-OS are further reflected in its open-set performance which, at best, is equal to the benchmark but otherwise shows comparatively low CCR throughout all FPR values compared to all other losses. On the negative samples, both, Cos-OS and Norm-OS have their OSCR curves drop below the benchmark at very high FPR values of roughly 4×10^{-1} and 10^{-1}, respectively. However, on the unknowns both losses consistently and clearly achieve higher CCR@FPR than the OS loss, with Norm-OS even extending further to the left and achieving the highest CCR of all losses at FPR $= 10^{-2}$.

Cos-EOS and Arc-EOS show similar behavior on the negatives and unknowns, where their OSCR curves are higher than the OSCR curves for OS and Norm-EOS, but drop below them at FPR values at or slightly above 10^{-2}. Norm-EOS, however, achieves consistently better or equal performance compared to the benchmark on the negatives and almost identical performance on the unknowns, except its increased closed-set accuracy.

Protocol 3 The closed-set accuracies on protocol P_3 for the Margin-OS losses is among the highest out of all losses, with Cos-OS having the highest accuracy of 0.80. The accuracies of Arc-OS and Norm-OS on the unknowns is roughly 0.79 and 0.78, respectively. All of which are higher than the Margin-EOS closed-set accuracies of about 0.77.

Not only does Cos-OS achieve the highest closed-set accuracy, but it also achieves the highest CCR at FPR $= 10^{-1}$ on the unknowns, namely, 0.60. As is to be expected, because the negatives and unknowns share many visual features, we observe almost identical OSCR curves for all Margin-OS losses and Margin-EOS between the negatives and unknowns. Interestingly, even between the two groups the OSCR curves behave very similarly. For Cos-OS, Arc-OS, Cos-EOS, and Arc-EOS the CCR@FPR values are above the OS benchmark for FPR values from roughly 3×10^{-2} to 1, from where the CCR steeply drops to values below 0.2. It is also noteworthy to highlight that even though the zero-margin versions of Margin-OS and Margin-EOS slightly lag behind the respective margin-based losses for high FPR values, for low FPR values they clearly achieve higher CCR@FPR values. In particular, Norm-EOS achieves largely slightly better performance compared to the OS benchmark, but reaches much lower FPR values, albeit at very low CCR below 0.2. Norm-OS and, in particular, all Margin-EOS losses extend to very small FPR values of down to 2×10^{-4}, thus achieving significantly lower FPR values compared to the benchmark.

Angle Distributions Figure 6.8 shows that Cos-OS shifts the angle distribution of the known samples towards the left with a very clear peak of angles below $\frac{\pi}{8}$ compared to the zero-margin version Norm-OS on all protocols. On protocols P_2 and P_3 where Arc-OS converged, it shows similar distributions of the knowns. Norm-OS also learns slightly smaller angles than the OS loss for known samples, but overall the distribution is not as wide as for Cos-OS and Arc-OS. As is to be expected, we can also see that the angular separation between knowns and negatives/unknowns becomes less clear with increasing difficulty of the protocols for all Margin-OS losses and the benchmark. Generally, the distributions of the negatives and unknowns overlap almost perfectly for all losses on all protocols. Similar to the SFN-Margin losses, the distribution of angles of the known classes are heavily right-skewed and the distributions of the margin-based losses show higher overlaps for very small angles on all protocols compared to Norm-OS or OS.

The Margin-EOS losses, which exclusively separate the classes based on the angles show much stronger separation of the angles. Since these losses use HFN, the probability curves can be expected to be accurate since both assumptions are fulfilled. Cos-EOS and Arc-EOS learn distributions for the knowns which peak at around $\frac{\pi}{8}$, which is where the probability curves reach 1 during training. This illustrates the influence of the margin, as Norm-EOS does not achieve

similarly small angles for the knowns. Analogous to the SFN-Margin and Margin-OS losses, the losses that apply a margin tend to skew the distributions of negatives and unknowns to the left and achieve an increased overlap in the distributions for small values. In contrast to the SFN-Margin and Margin-OS losses, as we can see that for all Margin-EOS losses the minimum angle for negative samples to any class is heavily concentrated around $\frac{\pi}{2}$, which is equal the average angle. This implies that the maximum angle of the negatives is $\frac{\pi}{2}$ or slightly larger as well, which is also what we observe empirically. Interestingly, for P_1 we can see that all Margin-EOS losses have distributions of unknown samples that don't overlap with the negatives as much as for the other protocols. We also verified that the pairwise angles between all class centers are at or slightly above $\frac{\pi}{2}$. From these observations we know that the Margin-EOS losses learn $C + 1$ class centers that are close to — but not exactly — mutually orthogonal.

Summary Arc-OS shows general signs of training instabilities and even fails to converge on P_1. With exception of Arc-OS all Margin-OS and Margin-EOS losses, notably including the zero-margin versions, achieve significantly higher closed-set accuracies compared to the benchmarks on all protocols. The closed-set accuracies are generally higher for losses that impose a margin, with the exception of Norm-OS on P_2, which achieves the highest closed-set accuracies out of all losses. The CCR values at low FPR values, however, are generally worse compared to the benchmarks. This effect seems stronger the more difficult the OSC task becomes.

While zero-margin losses Norm-OS and Norm-EOS generally perform worse on P_1 compared to the benchmarks, they generally achieve similar or better performances to the benchmarks on the protocols P_2 and P_3. They also achieve much lower FPR values on P_3 along with Cos-EOS and Arc-EOS, compared to the benchmarks.

The margin-based losses Cos-OS, Arc-OS, Cos-EOS, and Arc-EOS achieve lower angles on the known samples than losses without margins, but their distributions of the knowns become wider. This leads to them being heavily right-skewed and reaching angles up to $\frac{\pi}{2}$. Notably, the margin-based approaches also skew the distributions of the negatives and unknowns to the left, such that they achieve angles that are as low as the smallest angles for the knowns. The Margin-EOS losses achieve a stronger angular separation between knowns and negatives/unknowns, with the latter ones being clustered such that their angles to all class centers are around $\frac{\pi}{2}$.

Table 6.2: IMAGENET OPEN-SET PROTOCOL RESULTS. *The tables in (a), (b), and (c) depict the performances on the unknown samples on protocols P_1, P_2, and P_3 respectively. Each table shows the CCR@FPR for all losses as well as the closed-set accuracy (Acc). Losses are grouped into: benchmarks, SNF Margin, Margin-OS, and Margin-EOS. For each column, the best performance is highlighted in* **blue**, *and the best per group is underlined. Empty cells indicate that the respective FPR was not reached.*

(a) Protocol 1

Group	Loss	CCR@FPR			Acc
		10^{-3}	10^{-2}	10^{-1}	1
Benchmarks	Softmax	0.1772	0.4048	0.5855	0.6726
	EOS	0.2648	0.4816	0.6643	0.6903
	Objectosphere	**0.3095**	0.5214	0.6666	0.6843
SFN Margin	SFN-Norm	0.1674	0.3814	0.5562	0.6719
	SFN-CosFace	0.1802	0.4729	0.6583	**0.7095**
	SFN-ArcFace	0.0503	0.5005	0.6809	0.6945
Margin-OS	Norm-OS	0.2897	0.4986	0.6612	0.7029
	Cos-OS	0.1803	0.5440	**0.6952**	0.7067
	Arc-OS	0.0000	0.0000	0.0009	0.0126
Margin-EOS	Norm-EOS	0.2172	0.4336	0.6436	0.6950
	Cos-EOS	0.2531	0.5278	0.6653	0.7060
	Arc-EOS	0.1386	**0.5472**	0.6733	0.7083

(b) Protocol 2

Group	Loss	CCR@FPR			Acc
		10^{-3}	10^{-2}	10^{-1}	1
Benchmarks	Softmax	0.0280	0.1160	0.3673	0.6260
	EOS		0.1073	0.4013	0.6253
	Objectosphere	0.0347	0.1607	0.4093	0.6320
SFN Margin	SFN-Norm	0.0527	0.1220	0.3267	0.6213
	SFN-CosFace		0.1100	0.4020	0.6653
	SFN-ArcFace	0.0407	0.1493	0.4220	0.6733
Margin-OS	Norm-OS	**0.0660**	0.1853	0.4420	**0.6800**
	Cos-OS	0.0573	**0.2080**	0.4473	0.6647
	Arc-OS		0.1173	0.4133	0.6273
Margin-EOS	Norm-EOS	0.0353	0.1647	0.4073	0.6573
	Cos-EOS	0.0127	0.1553	**0.4793**	0.6740
	Arc-EOS	0.0027	0.1080	0.4627	0.6733

(c) Protocol 3

Group	Loss	CCR@FPR			Acc
		10^{-3}	10^{-2}	10^{-1}	1
Benchmarks	Softmax		0.2238	0.5163	0.7574
	EOS		0.2517	0.5428	0.7630
	Objectosphere		0.2562	0.5294	0.7482
SFN Margin	SFN-Norm	**0.0809**	0.2351	0.5275	0.7597
	SFN-CosFace		0.0805	0.5501	0.7894
	SFN-ArcFace		0.0623	0.5536	0.7901
Margin-OS	Norm-OS	0.0411	0.2185	0.5832	0.7768
	Cos-OS		0.0877	**0.6044**	**0.7952**
	Arc-OS		0.1159	0.5874	0.7858
Margin-EOS	Norm-EOS	0.0531	**0.2596**	0.5576	0.7674
	Cos-EOS	0.0097	0.1417	0.5909	0.7713
	Arc-EOS	0.0033	0.0838	0.5932	0.7764

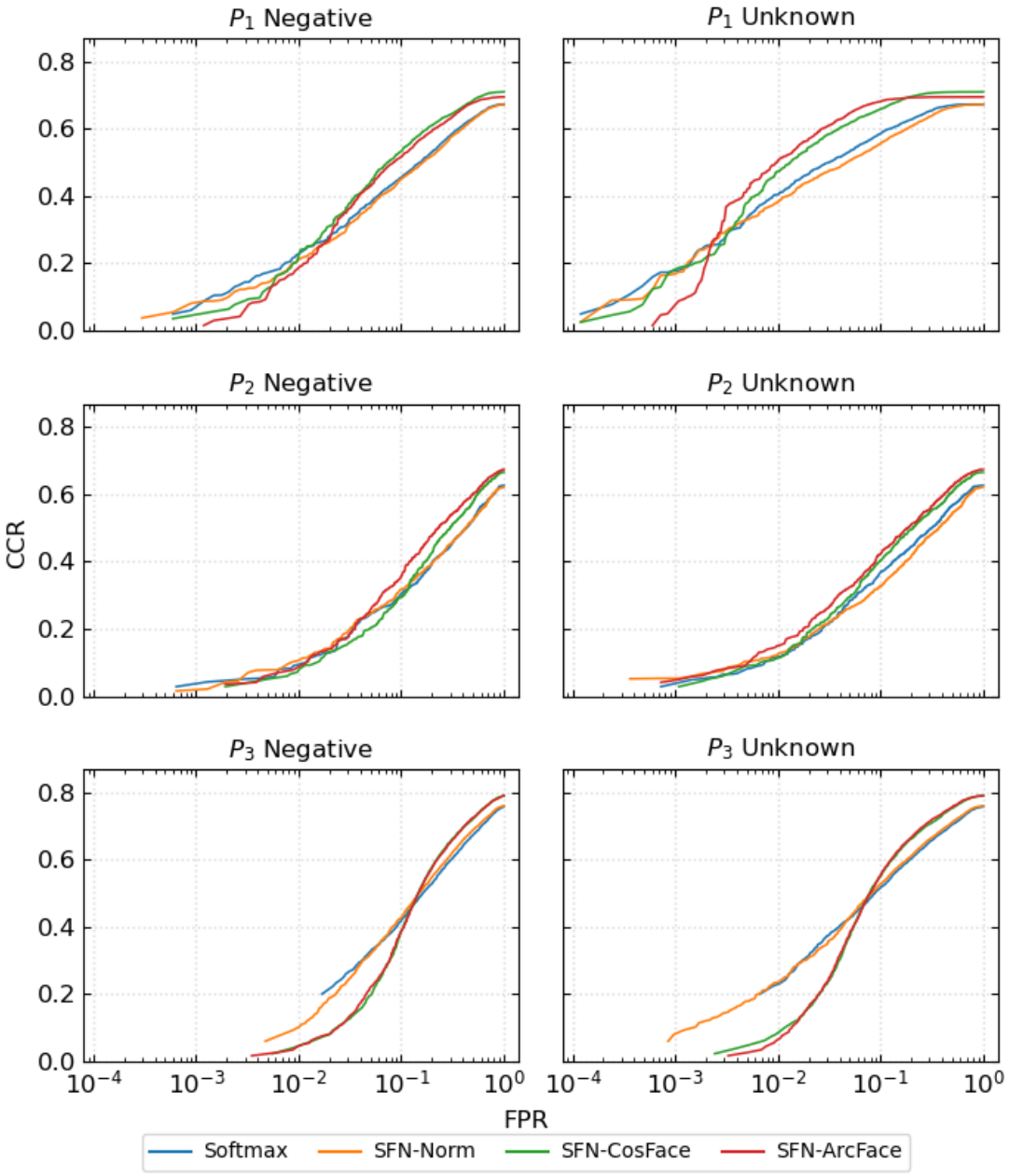


Figure 6.5: OSCR CURVES OF SFN-MARGIN LOSSES. *This figure shows the OSCR curves of the SFN-Margin losses for each ImageNet open-set protocol and for the negative and unknown tests samples respectively. The softmax loss is the benchmark.*

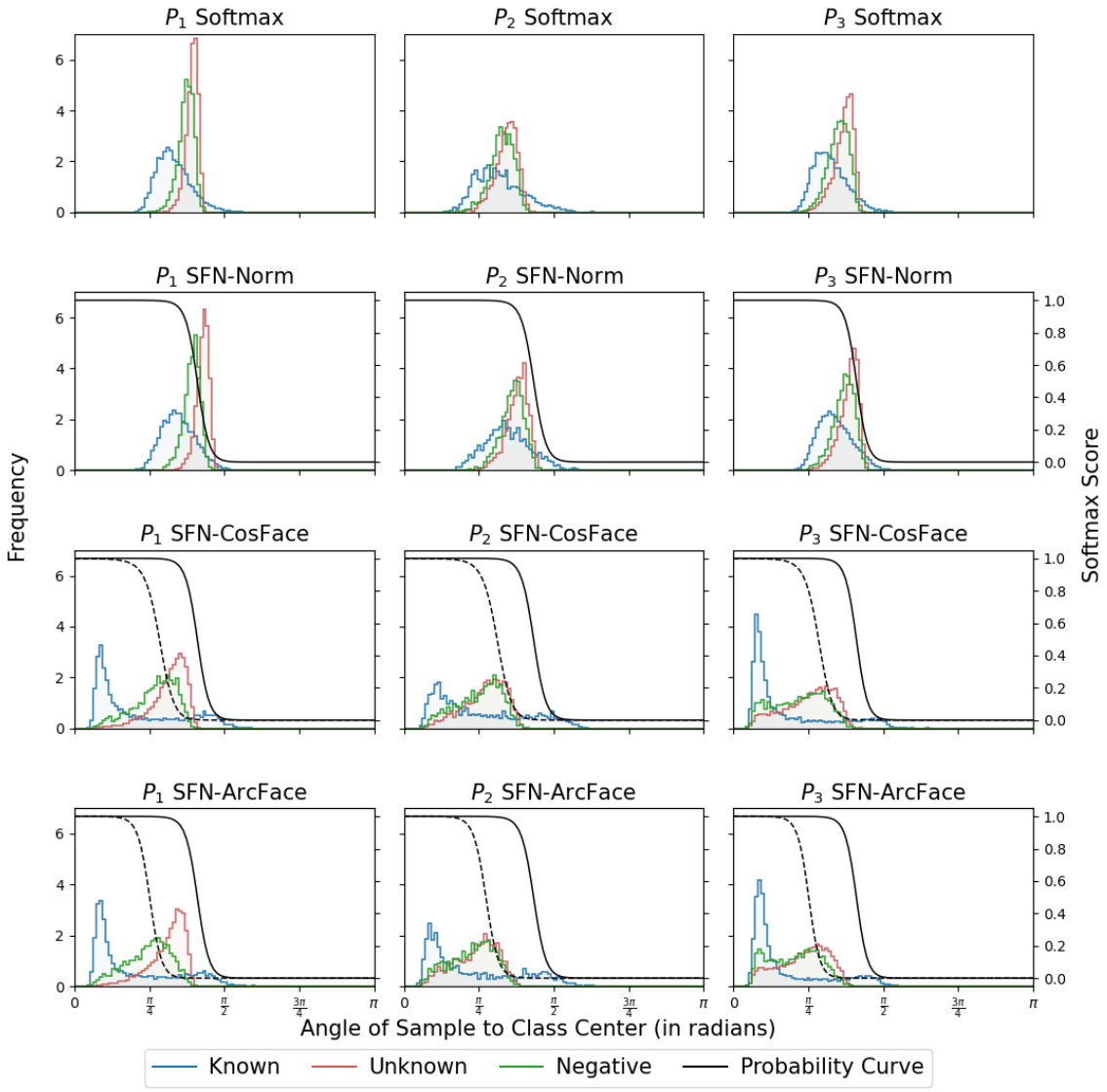


Figure 6.6: ANGLE DISTRIBUTIONS WITH PROBABILITY CURVES OF SFN-MARGIN LOSSES. *This figure shows the distributions of the angles (in radians) of known, negative, and unknown test samples of the SFN-Margin losses and the softmax loss as benchmark. The known samples depict their angles to the ground-truth class center ($\theta_{i,\mathbf{y}_i}$). The negative and unknowns depict theeir angles to the closest class center ($\min_{c \in \mathcal{C}}\{\theta_{i,c}\}$). For each loss with normalized features and weights we superimpose the probability curves as a function of the angle (solid black line) with consideration of the scaling factor s. For margin-based losses we add the probability curve with consideration of the respective margin (dashed black line).*

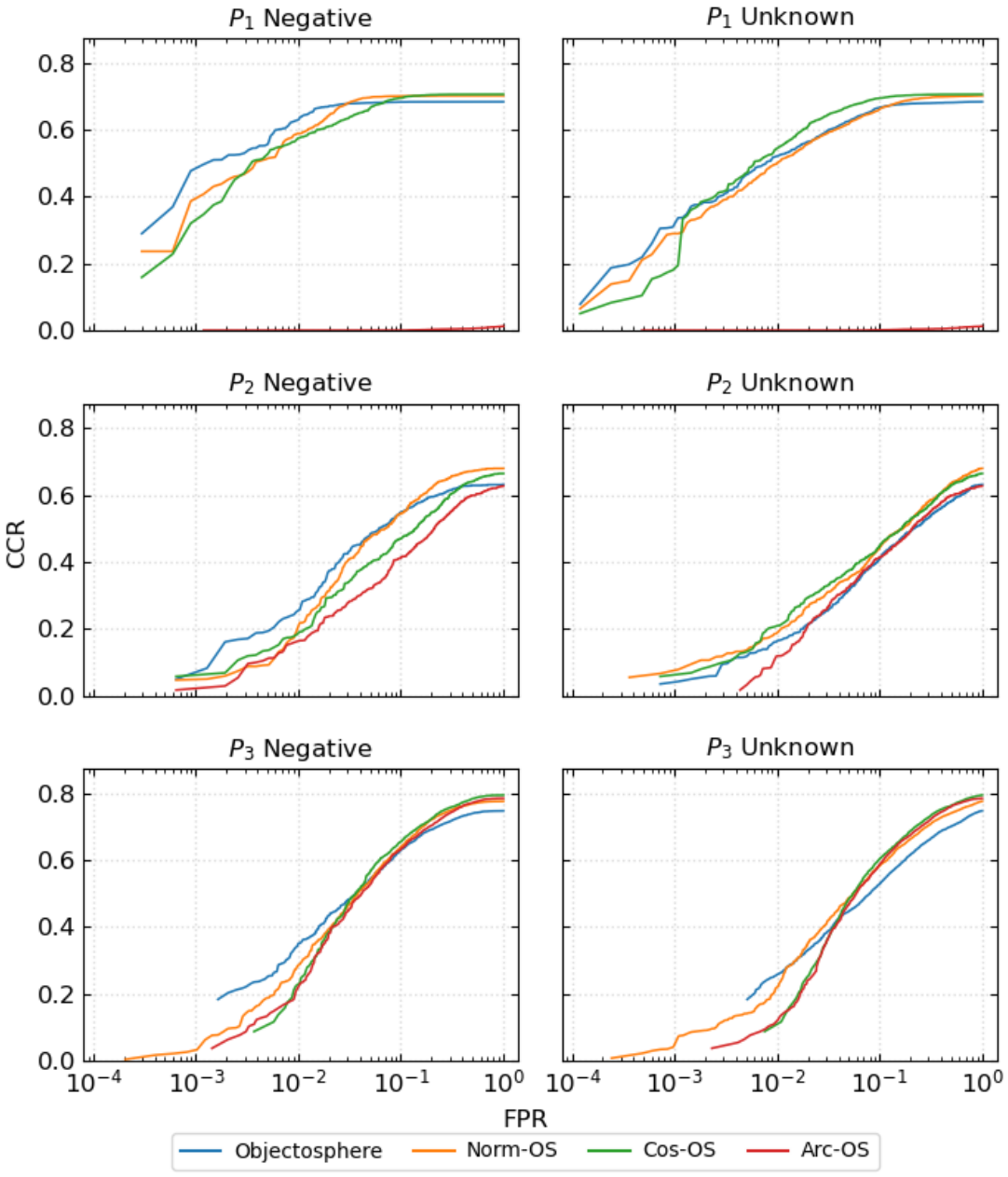


Figure 6.7: OSCR CURVES OF MARGIN-OS LOSSES. *This figure shows the OSCR curves of the Margin-OS losses for each ImageNet open-set protocol and for the negative and unknown tests samples respectively. The OS loss is the benchmark.*

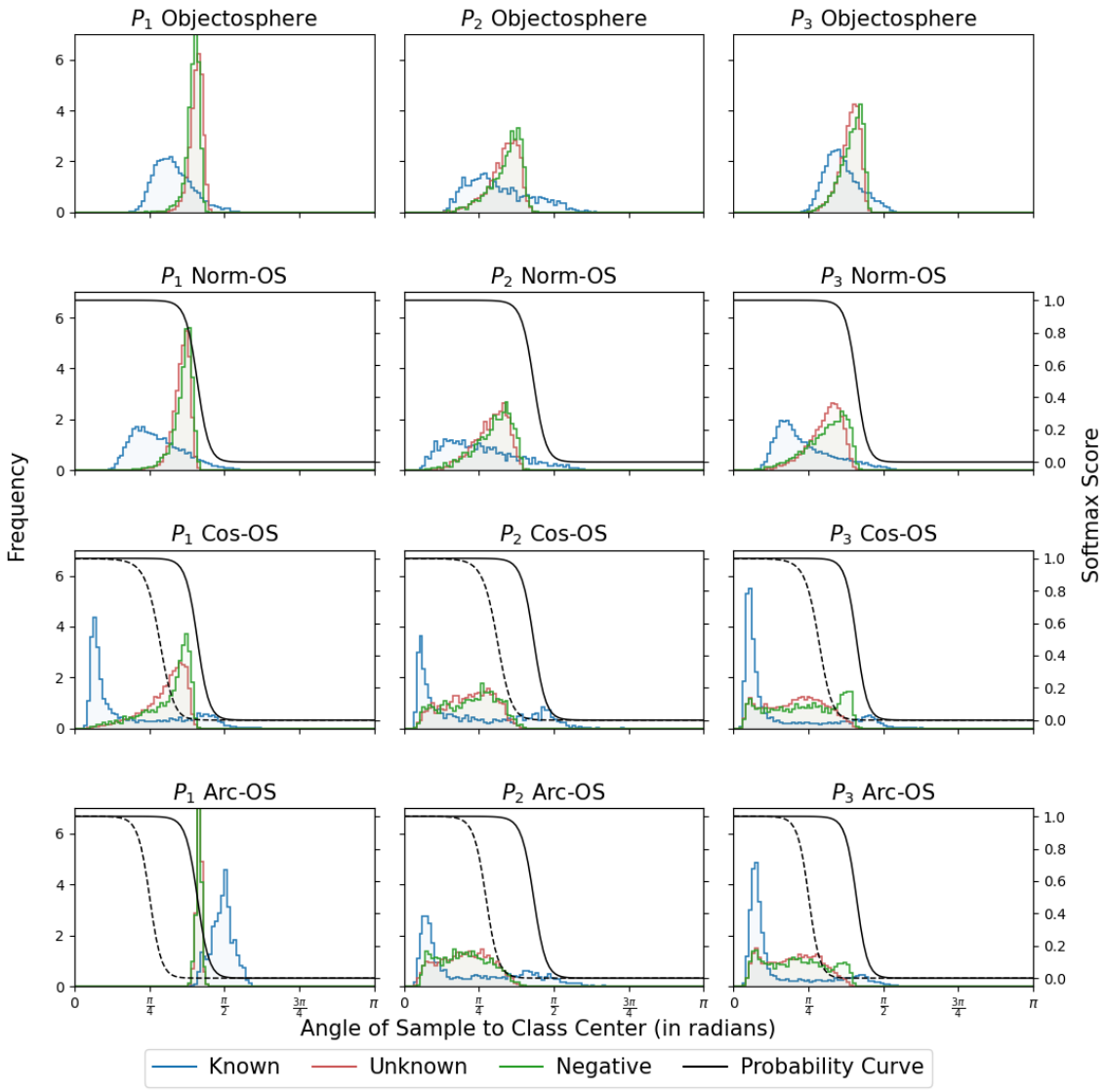


Figure 6.8: ANGLE DISTRIBUTIONS WITH PROBABILITY CURVES OF MARGIN-OS LOSSES. *This figure shows the distributions of the angles (in radians) of known, negative, and unknown test samples of the Margin-OS losses and the OS loss as benchmark. The known samples depict their angles to the ground-truth class center ($\theta_{i,\mathbf{y}_i}$). The negative and unknowns depict theeir angles to the closest class center ($\min_{c\in\mathcal{C}}\{\theta_{i,c}\}$). For each loss with normalized features and weights we superimpose the probability curves as a function of the angle (solid black line) with consideration of the scaling factor s. For margin-based losses we add the probability curve with consideration of the respective margin (dashed black line).*

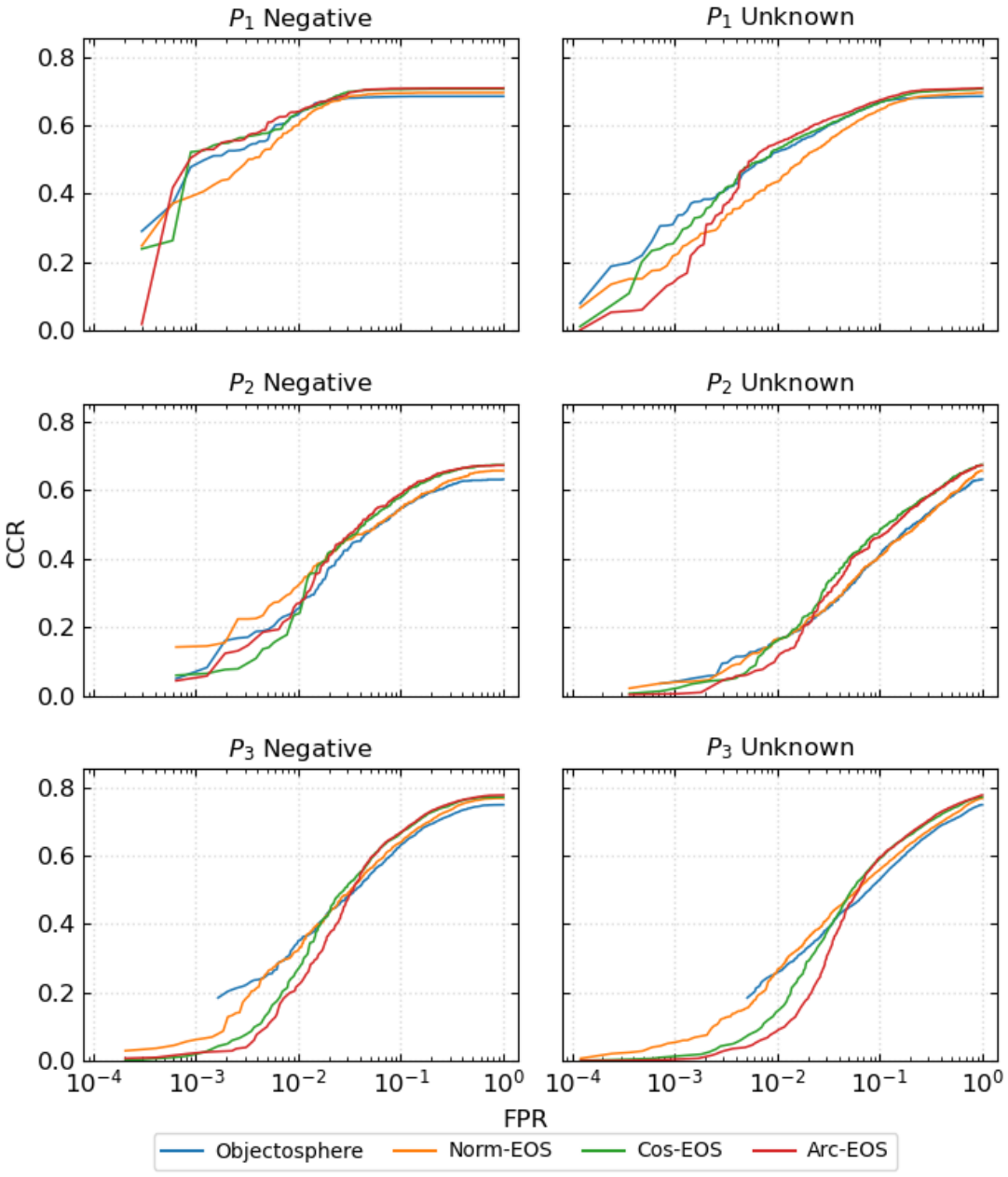


Figure 6.9: OSCR CURVES OF MARGIN-EOS LOSSES. *This figure shows the OSCR curves of the Margin-EOS losses for each ImageNet open-set protocol and for the negative and unknown tests samples respectively. The OS loss is the benchmark.*

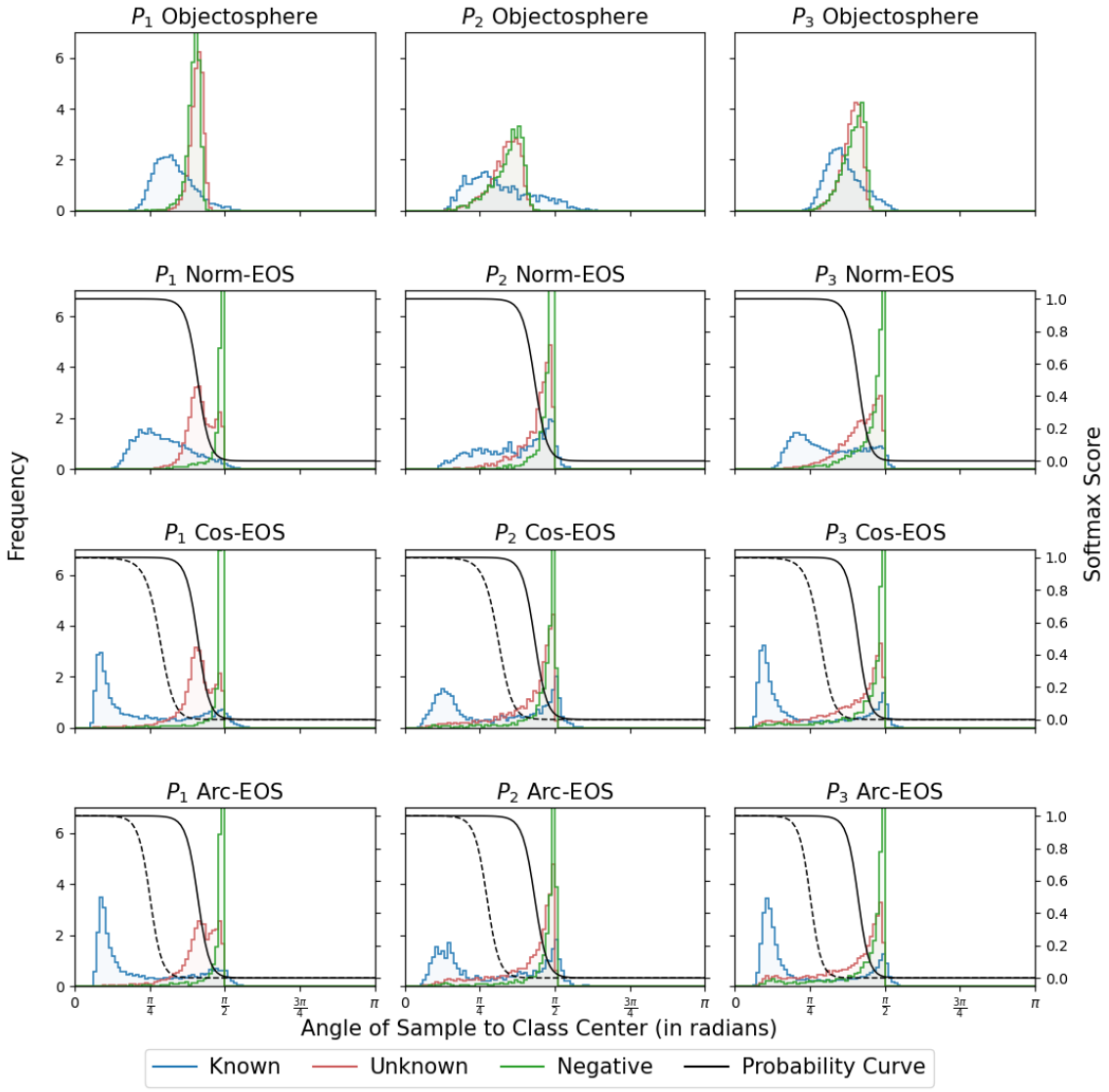


Figure 6.10: ANGLE DISTRIBUTIONS WITH PROBABILITY CURVES OF MARGIN-EOS LOSSES. *This figure shows the distributions of the angles (in radians) of known, negative, and unknown test samples of the Margin-EOS losses and the OS loss as benchmark. The known samples depict their angles to the ground-truth class center ($\theta_{i,\mathbf{y}_i}$). The negative and unknowns depict theeir angles to the closest class center ($\min_{c\in\mathcal{C}}\{\theta_{i,c}\}$). For each loss with normalized features and weights we superimpose the probability curves as a function of the angle (solid black line) with consideration of the scaling factor s. For margin-based losses we add the probability curve with consideration of the respective margin (dashed black line).*

Chapter 7

Discussion

In this section we discuss and interpret the results from our experiments in order to answer our research questions with consideration of the limitations of our work. Even though our research questions do not specifically ask for the effect of feature and weight normalizations, we are able to draw conclusions about this effect as well by comparing the zero-margin versions to the benchmarks.

Probability Curves Before addressing the research questions we want to highlight the significance of the probability curves for our interpretation. The probability curves highlight a correspondence from the angles to softmax scores. While this is not a one-to-one correspondence for approaches that employ SNF, it still provides intuition on the softmax score that is likely to be achieved for any specific angle. It is also important to keep in mind that the two probability curves of margin-based approaches are to be interpreted differently. The one shifted to the left (dashed) provides the supervision feedback during training, while the one to the right (solid) is used to compute predictions at test time, since we cannot apply a margin at test time. This results in significantly more samples achieving maximal softmax scores of 1 at test time than during training. This is a fundamental problem when thresholding the softmax scores, because each sample with score 1 will always be classified as known and thus result in higher false positive rates.

7.1 Effect of the Margin without Negative Samples (RQ1)

Research question RQ1 asks: *What effect do margins from margin-based loss functions have on the open-set (RQ1a) and closed-set (RQ1b) performance of an OSC task, when trained without negative samples?*

Effect on Closed-set Performance (RQ1a) We clearly see a consistently increased closed-set performance from SFN-CosFace and SFN-ArcFace over softmax throughout all protocols. Since the zero-margin version, SFN-Norm, shows comparable performances to softmax on all protocols, it is safe to conclude that the increased closed-set performance is a result of the respective margins. As such, subquestion RQ1a can be clearly answered as we find a positive effect (increased accuracy) of imposing a margin on the closed-set performance. However, we do not see a clear indication as to whether the additive cosine or angular margin is better suited, since the results vary for each protocol with no clear trend.

This result is not surprising because the margins are designed to move known samples further away from the decision boundaries and closer to the respective class center. We can also see that the normalization of the features and class centers does not show any effect on the ability of the classifiers to correctly classify samples on a closed-set classification task.

Effect on Open-Set Performance (RQ1b) Analyzing the effect of the margin on the open-set performance is less clear. On P_1 we can see that imposing a margin helps to separate between knowns and negatives and unknowns, respectively. The more we increase the threshold (and lower the FPR) on SFN-CosFace and SFN-ArcFace, the worse their relative performance becomes compared to softmax and SFN-Norm. Crucially, since the performance deteriorates relative to SFN-Norm, it is likely that the decreasing open-set performance is a result of the margin.

Ultimately, we do not know for certain what causes this effect. It is likely a result of the margin drawing negative and unknown samples towards class centers and increasing the dispersion in the angles of known samples. This can have two effects: (1) samples with very low angles become false positives or (2) known samples achieve lower softmax scores and become rejected more easily. While both this would explain the drop in CCR, the former would also coincide with stagnating FPR values at comparatively high values, which is not what we observe when comparing the margin-based losses to SFN-Norm.

Additionally, in the appendix we provide the OSCR curves (Figure A.1) for the SFN-Margin losses under evaluation via thresholding of the logits, to analyze if the decrease in CCR is a result of the scores reaching 1 and becoming difficult to threshold. Only for P_3 can we see a decrease in FPR values, which would support this argument, while for all other protocols, the margin does not lead to any change in the FPR. This indicates that the scores are likely not too large.

Because the CCR only drops for high thresholds and is clearly higher for low thresholds, we believe that the margin does not counterintuitively lead to more misclassifications of known samples. Instead, we hypothesize that the decrease in CCR is in fact a result of larger angles for many known samples, which in turn lead to scores that are strictly smaller than one, but not by much. This would result in a drop of CCR values for large thresholds that leads to knowns not being recognized as such while keeping the FPR values largely unaffected. The angle distributions of the known samples along with the corresponding probability curve without a margin (Figure 6.6) supports this interpretation as many scores are likely to not reach scores less than 1. This can somewhat be counteracted by the SFN adjusting to learn larger magnitudes but the effect cannot be mitigated entirely, since the cosine of the angle and the feature magnitude are multiplied, meaning that cosine values around 0 will not be able to counteracted. Further analysis of other metrics, such as the false negative rate is needed. We observe that with increasing open-set difficulty of the protocols, the downsides of the margin also become more pronounced, as the underperformance of SFN-CosFace and SFN-ArcFace compared to SFN-Norm on P_3 is much stronger compared to protocols P_1 and P_2.

In conclusion, we can answer research question RQ1b as follows: The effect of the margin on open-set performance is positive when applying low thresholds which results in higher CCR. However, for higher thresholds, the effect becomes negative, as the margin-based methods presumably fail to successfully identify known samples which leads to a high rejection rate, while still achieving low false positive rates. The downsides of applying a margin seem to become more pronounced the harder the OSC task is.

7.2 Effect of the Margin with Negative Samples (RQ2)

Research question RQ2 asks: *What effect do margins from margin-based loss functions have on the open-set performance (RQ2a) and closed-set performance (RQ2b) of an OSC task, when combined with EOS and*

OS to incorporate negative samples during training? First and foremost we need to highlight that, while we propose two generic loss functions that combine the margin-based losses CosFace and ArcFace with EOS and OS, we cannot extrapolate the effect we observe to any loss that modifies EOS and OS to impose margins between known classes. The effects we observe seem largely similar for both of our losses but this does not guarantee that losses can exist where the effect of the margin is different. As such we answer these questions with respect to our proposed loss functions.

Effect on Closed-set Performance (RQ2a) Similar to the closed-set performance of the SFN-Margin losses, the Margin-OS and Margin-EOS losses show a clear and strong tendency to achieve increased closed-set accuracies on all protocols over the benchmarks. In contrast to the SFN-Margin losses, the isolated effect of the margin for Margin-OS and Margin-EOS is not as strong when compared to the respective zero-margin losses. For the Cos-OS losses on protocol P_2 the isolated effect of the margin is even negative. This suggests that, while the effect is mostly positive, it is not as strong when incorporating negative samples during training. It appears that forcing the networks to discriminate between known classes only based on the angle (when normalizing feature and weights) shows a stronger positive effect than imposing a margin. Comparing Margin-OS and Margin-EOS in terms of their closed-set performance, we cannot see a strong difference between the two, except that Margin-OS seems to have a slight edge over Margin-EOS. As such, there is no clear preference on how to deal with negatives in a closed-set task, *i. e.*, discriminating between knowns and negatives/unknowns based on the angle or feature magnitude.

We can answer research question RQ2a as follows: The effect of the margin on the closed-set accuracy, when the network is trained with negative samples, is mostly positive but small and possibly insignificant in terms of its effect size. Forcing the networks to distribute deep features around a hypersphere and only discriminating the known classes based on the angle, by normalizing features and weights, seems to show a stronger effect.

Effect on Open-Set Performance (RQ2b) Interestingly, the effect of the margin on the open-set performance seems to be largely unchanged when including negative training samples. Throughout all protocola, all loss functions show a steep decrease in CCR for high thresholds while showing similar performance in terms of FPR or even improving over the benchmarks. Thresholding the logits instead of the softmax scores does in fact yield slight improvements but does not fully remove — and thus does also not explain — the steep drop in CCR of the margin-based losses (see Figure A.2 and Figure A.3).

We also observe that Cos-OS, Arc-OS, Cos-EOS, and Arc-EOS increase the dispersion in the angles of the known samples. Norm-OS and Norm-EOS shows a similar effect but less strong. We suspect that, analogously to the SFN-Margin losses, the steep decrease in CCR values at thresholds close to 1 is the result of known samples not reaching softmax scores of 1 exactly. This leads to them being rejected while keeping reaching very low FPR values. This effect seems to be strongest on protocol P_3 but similar tendencies can be observed on the other protocols as well. Similarly to the SFN-Margin losses, we thus believe that the downsides of applying a margin become more pronounced with increased difficulty of the OSC task.

Since the Margin-EOS losses employ HFN, the probability curves in Figure 6.10 are expected to be relatively accurate, which suggest that many known samples achieve scores very close to 0. This provides strong evidence that this is causing known samples to be rejected with large thresholds since many do not reach scores of 1.

We answer research question RQ2b as follows: The effect of the margin on open-set performance positive when low thresholds are applied to the softmax scores. Imposing margins, however, seems to have a negative side effect that leads to the distribution of angles of known samples being heavily right-skewed which results in some known samples achieving small probability

scores for the ground-truth class and getting rejected at high thresholds. This negative side effect seems to become more pronounced the harder the OSC task is. Forcing the networks to discriminate between classes only based on the angles seems to explain most of the increased accuracy and CCR while circumventing the negative effects of imposing margins.

7.3 Limitations

Following we want to highlight individual remarks and limitations of our work:

- For each loss we explore two different margin types: the additive cosine margin and the additive angular margin. Because we do not optimize the margin parameters for each loss individually, we cannot directly compare the effects of these margins, as we do not know if both chosen parameters achieve equally optimal performance for the respective losses. The angular margin, however, shows instabilities during training of the Arc-OS loss, but we currently do not know what caused this. This issue is related to the fact that, due to time constraints, we do not quantify the uncertainty in the CCR@FPR values. This makes comparisons of CCR@FPR values very difficult as we do not know which differences are significant and which are likely a result of random chance.
- Since this is a comparative study and we choose training parameter such that they lead to reasonable results for most losses, we cannot exclude that certain losses might benefit from vastly different training parameters. For example, when analyzing the validation loss for all Margin-EOS losses we observe that they converge very fast (about 30 epochs) to a state from which not the other 90 epochs show little progress. This could be an indicator that the learning rate is too small. Also, training for a fixed number of epochs will favor losses that benefit from more iterations over losses that converge quickly and consequently run the risk of overfitting to the training data more.
- While we compute a lower bound on the scale parameter s as a function of the expected minimum posterior probability $\hat{\mathbf{p}}_{i,c}$, that is simpler to guess reasonably, it is ultimately still a hyperparameter that ideally should be tuned and experimented on. Considering our results it is possible that we chose $\hat{\mathbf{p}}_{i,c}$ too small to not obtain a scale factor s that is too large, and as a result ended up with s being too small.
- We observe, that the Margin-EOS losses learn deep features for the negatives for which the angles to all other class centers are clustered at $\frac{\pi}{2}$ with a skew towards smaller angles. This can be interpreted as an implicit background class center in the deep feature space. Interestingly, the pairwise angles between all class centers is also very close to $\frac{\pi}{2}$ but occasionally a bit larger. This indicates that Margin-EOS wants to learn $C+1$ mutually orthogonal vectors, which crucially requires $K = C + 1$. As such, the deep feature dimensionality of $K = C$ possibly leads to worse performance than could have been possible.
- We acknowledge that the decision to remove certain visually indistinguishable letters from our toy protocol is to a certain degree an arbitrary choice.

Chapter 8

Conclusion and Future Work

In this thesis we explore the effect that imposing a margin between deep features of known classes has on closed-set and open-set classification performance. This thesis started on the hypothesis that learning discriminative features can increase said closed-set and open-set performance. We analyze this effect for losses that are only trained on known samples and losses that incorporate negative samples during training. To analyze the former, we adapt the CosFace loss and ArcFace loss with a Ring loss regularizer to achieve SFN and obtain the SFN-CosFace and SFN-ArcFace losses. To analyze the latter, we propose two novel loss generic functions Margin-OS and Margin-EOS, that are combinations of the EOS and OS losses with the CosFace and ArcFace loss.

To achieve effective training with margins, we normalize the weights in the logit layer and employ SFN or HFN on the deep features. This forces the networks to discriminate known classes primarily based on the angle to the class centers. The Margin-OS losses discriminate between knowns and negatives/unknowns via the feature magnitude, learning the zero vector as deep feature for negative and unknown samples, while learning a feature magnitude s for known samples. The Margin-EOS losses discriminate between known and negative/unknown classes only via the angle, which lets us interpret the Margin-EOS losses as learning an implicit background class center for negative and unknown samples.

We train all networks on three ImageNet open-set protocols of varying degrees of difficulties. We find that a margin has a clear positive effect on the closed-set accuracy throughout all protocols and losses. However, the effect becomes smaller when training includes negative samples, where normalizing weights and features without imposing a margin shows a large positive effect on closed-set accuracy and explains most of the increased closed-set accuracy for the Margin-OS and Margin-EOS losses. The effect of the margin on open-set classification is less clear. We find that when applying low thresholds to the softmax scores, an open-set classifier can benefit from imposing a margin between classes as it leads to increased CCR. However, this effect only holds for relatively high FPR values and thresholds that are clearly smaller than 1. In fact, for thresholds close to 1 and consequently small FPR values, the margin has a negative impact on the CCR, as it leads to overproportionally many known samples being rejected compared to losses that do not impose a margin. This negative effect renders the margin-based losses useless for safety critical applications that require very few false positives.

Future Work To close the thesis off, we want to highlight potential considerations for future work:

- The scale parameter s is of great importance for any margin-based loss as these require either hard or soft feature normalization. While we do have a lower bound on this parameter, more research into the effect that the feature magnitude has on the performance of OSC

methods is needed. This might include derivations of better bounds that, for example, account for the margin parameter, or adaptions of our proposed losses to methods such as AdaCos (Zhang et al., 2019) which dynamically adapt the feature magnitude during training.

- While we observed the effect of the margin on closed-set and open-set classification performance, it would be interesting to further explore what causes these effects to potentially find ways to counteract the negative effect on open-set performance.
- Since the effect of the margin on closed-set classification is clearly positive, it would be interesting to combine the SFN-Margin losses with a background class approach, to learn a margin between known and negative/unknown samples.

Appendix A

Attachments

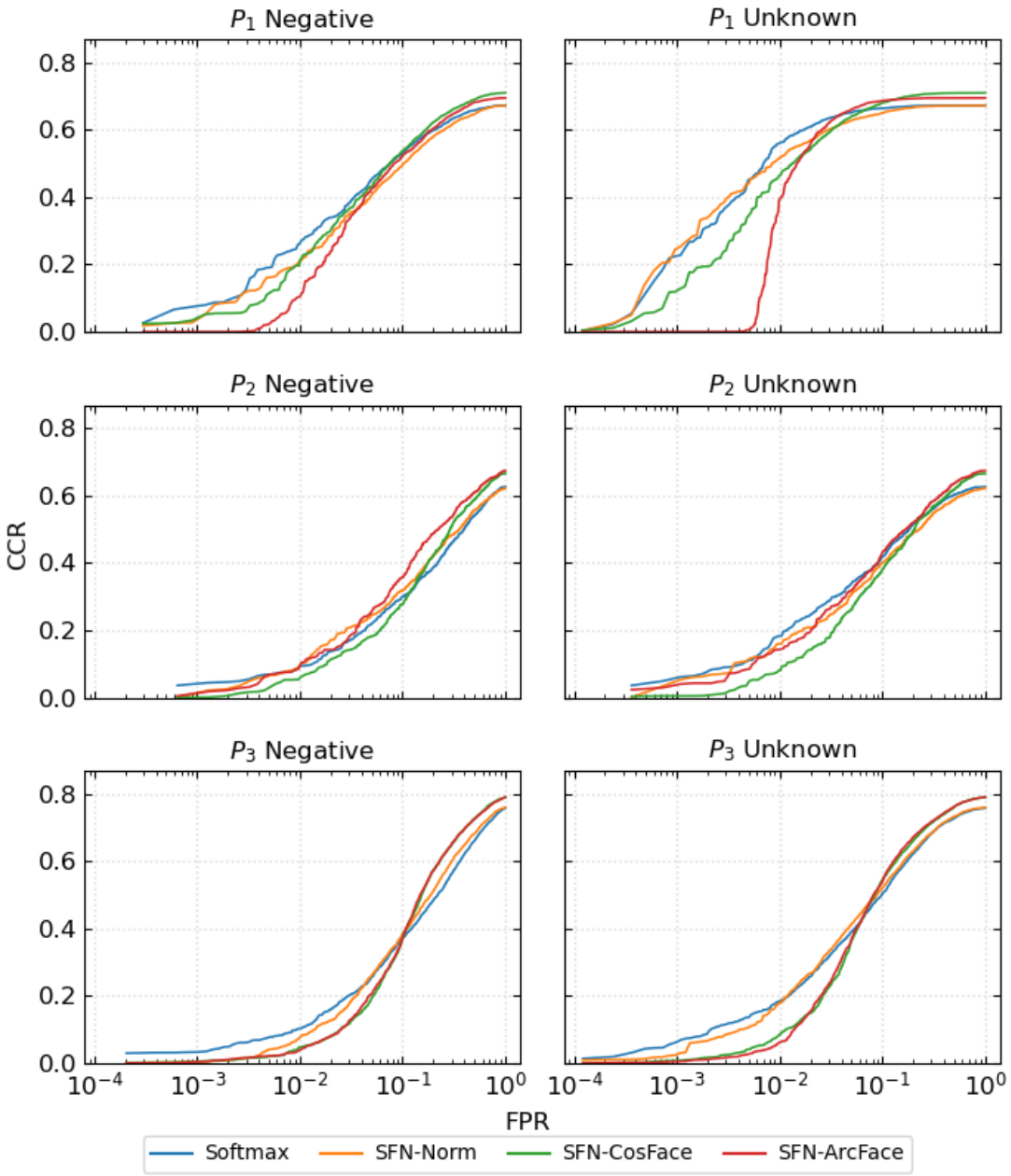


Figure A.1: OSCR Curves of SFN-Margin Losses for Logit Thresholding. *This figure shows the OSCR curves of the SFN-Margin losses for each ImageNet open-set protocol and for the negative and unknown tests samples respectively. In contrast to Figure 6.5, the curves are computed by thresholding the logits instead of the softmax scores. The softmax loss is the benchmark. These results are discussed in Chapter 7.*

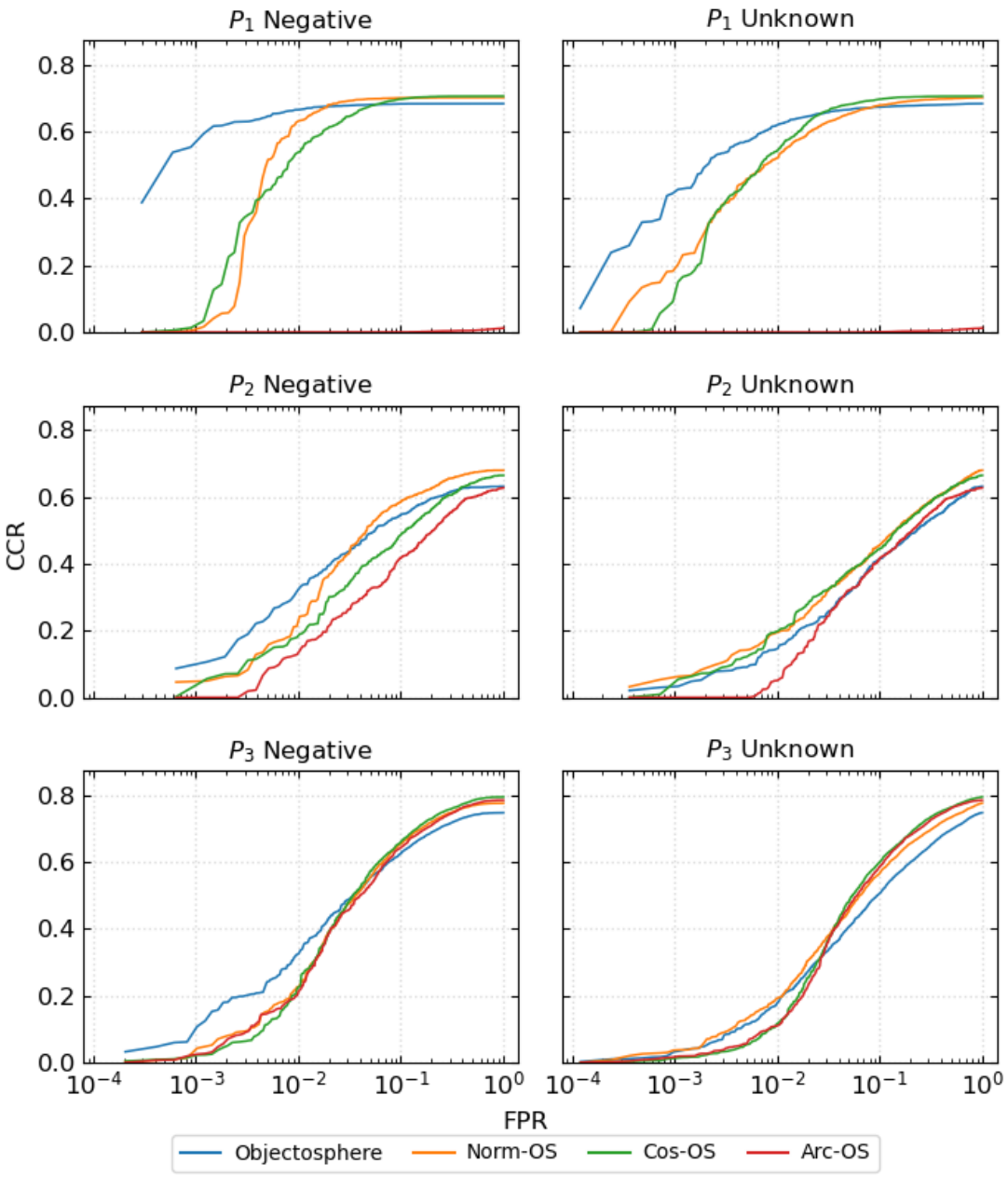


Figure A.2: OSCR CURVES OF MARGIN-OS LOSSES FOR LOGIT THRESHOLDING. *This figure shows the OSCR curves of the Margin-OS losses for each ImageNet open-set protocol and for the negative and unknown tests samples respectively. In contrast to Figure 6.7, the curves are computed by thresholding the logits instead of the softmax scores. The OS loss is the benchmark. These results are discussed in Chapter 7.*

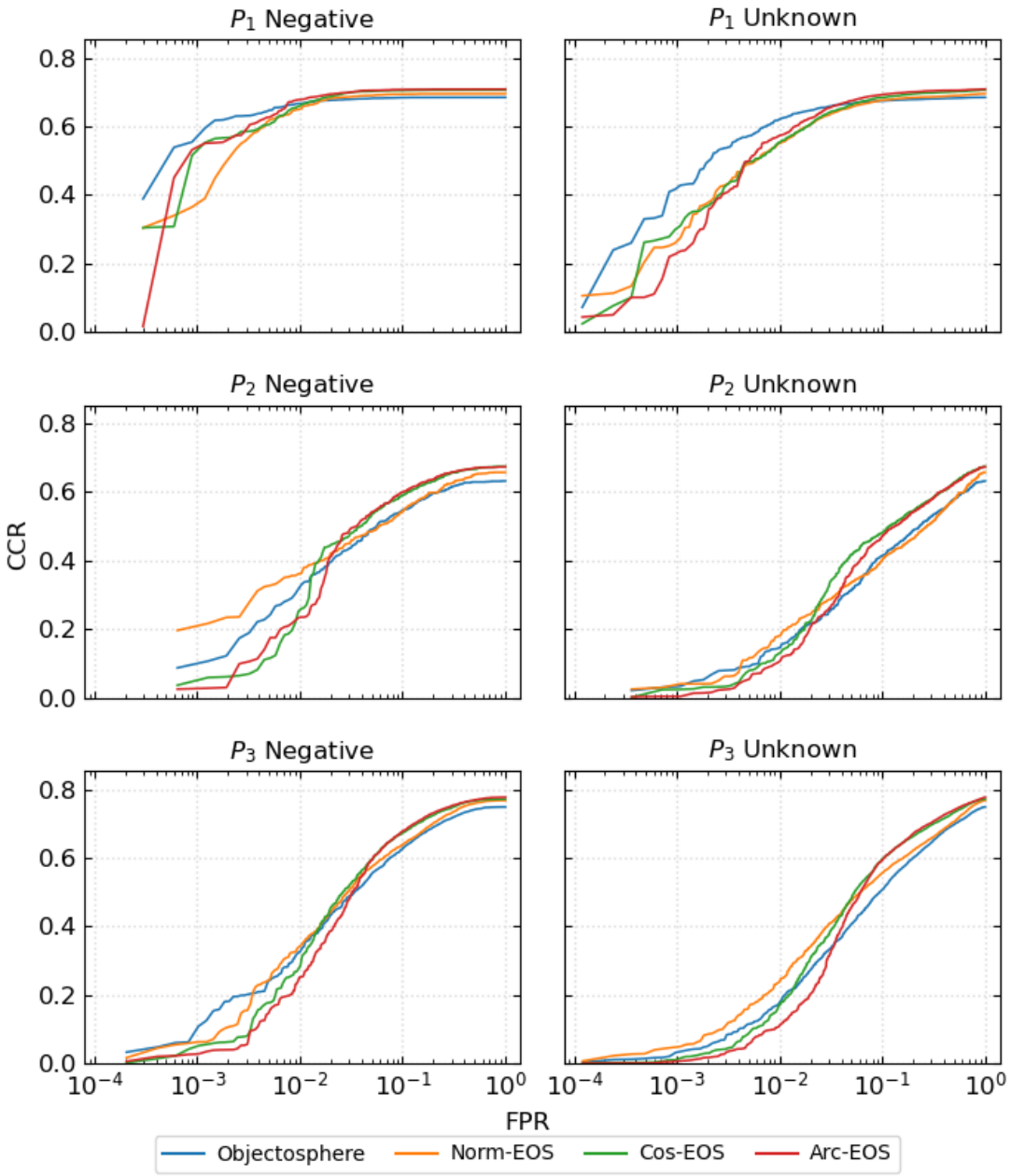


Figure A.3: OSCR Curves of Margin-EOS Losses for Logit Thresholding. *This figure shows the OSCR curves of the Margin-EOS losses for each ImageNet open-set protocol and for the negative and unknown tests samples respectively. In contrast to Figure 6.9, the curves are computed by thresholding the logits instead of the softmax scores. The OS loss is the benchmark. These results are discussed in Chapter 7.*

Lists of Symbols

Sets of class labels

$\mathcal{Y} \subset \mathbb{N}$	Set of all infinitely possible class labels
$\mathcal{C} = \{1, \dots, C\} \subset \mathcal{Y}$	Set of finitely many known classes
C	Cardinality of set $\mathcal{C}$, *i. e.*, $C = \lvert\mathcal{C}\rvert$
$\mathcal{M} = \mathcal{Y} \setminus \mathcal{C}$	Infinite set of mixed unknown classes
$\mathcal{N} \subset \mathcal{M}$	Finite set of negative classes
$\mathcal{U} = \mathcal{M} \setminus \mathcal{N}$	Infinite set of unknown classes

Datasets

$\mathcal{D}_{\mathcal{C}}^{\text{train}}$	Train dataset of known samples
$\mathcal{D}_{\mathcal{C}}^{\text{test}}$	Test dataset of known samples
$\mathcal{D}_{\mathcal{N}}^{\text{train}}$	Train dataset of negative samples
$\mathcal{D}_{\mathcal{N}}^{\text{test}}$	Test dataset of negative samples
$\mathcal{D}_{\mathcal{U}}^{\text{test}}$	Test dataset of unknown samples
$\mathcal{D}_{\mathcal{M}}^{\text{test}} = \mathcal{D}_{\mathcal{U}}^{\text{test}} \cup \mathcal{D}_{\mathcal{N}}^{\text{test}}$	Test dataset of mixed unknown samples
$\mathcal{D}^{\text{train}} = \mathcal{D}_{\mathcal{C}}^{\text{train}} \cup \mathcal{D}_{\mathcal{N}}^{\text{train}}$	Train dataset
$\mathcal{D}^{\text{test}} = \mathcal{D}_{\mathcal{C}}^{\text{test}} \cup \mathcal{D}_{\mathcal{M}}^{\text{test}}$	Test dataset
$N = \left\lvert\mathcal{D}^{\text{train}}\right\rvert$	Size of the training data

Neural network components

$\mathbf{x}_i$	Input data point or sample with index $i \in \{1, \dots, N\}$
$\mathbf{y}_i \in \mathcal{C}$	Ground-truth class label for sample $\mathbf{x}_i$
$\mathbf{t}_i \in \mathbb{R}$	Target for sample $\mathbf{x}_i$, typically one-hot encoding of $\mathbf{y}_i$
$\boldsymbol{\phi}_i \in \mathbb{R}^K$	(Deep) feature representation (embedding) of $\mathbf{x}_i$
$K \in \mathbb{N}$	Dimensionality of deep feature representations
B	Neural network backbone or feature extractor: $B(\mathbf{x}_i) = \boldsymbol{\phi}_i$
$\mathbf{z}_i \in \mathbb{R}^C$	Logit vector of sample $\mathbf{x}_i$
L	Logit function mapping features $\boldsymbol{\phi}_i$ to logits $\mathbf{z}_i$: $L(\boldsymbol{\phi}_i) = \mathbf{z}_i$
$\mathbf{W} \in \mathbb{R}^{K \times C}$	Weight matrix of the logit lunction
$\mathbf{W}_c \in \mathbb{R}^K$	c-th column vector of $\mathbf{W}$ representing the class center of class $c \in \mathcal{C}$
$\theta_{i,c}$	Angle between deep feature $\boldsymbol{\phi}_i$ and class center $\mathbf{W}_c$
$\mathbf{p}_i \in \mathbb{R}^C$	Softmax scores/probability distribution of sample $\mathbf{x}_i$
σ	Softmax function: $\sigma(\mathbf{z}_i) = \mathbf{p}_i$
H	Neural network head: $H(\boldsymbol{\phi}_i) = \sigma(L(\boldsymbol{\phi}_i)) = \mathbf{p}_i$
$\tau \in [0, 1]$	Softmax score (probability) threshold

List of Figures

List of Tables

Bibliography

Acharya, S. and Gyawali, P. (2016). Devanagari Handwritten Character Dataset. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XS53.

Bhoumik, A. (2021). Open-set classification on ImageNet. Master's thesis, University of Zurich.

Bisgin, H., Palechor, A., Suter, M., and Günther, M. (2023). Large-scale evaluation of open-set image classification techniques. *Journal of Machine Learning Research (JMLR)*. **under submission**.

Boult, T. E., Cruz, S., Dhamija, A., Gunther, M., Henrydoss, J., and Scheirer, W. (2019). Learning and the unknown: Surveying steps toward open world recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9801–9807.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. (2017). Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dhamija, A. R. (2022). *Five Roads to Open-set Recognition*. PhD thesis, University of Colorado Colorado Springs.

Dhamija, A. R., Günther, M., and Boult, T. E. (2018). Reducing network agnostophobia. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9157–9168.

Du, H., Shi, H., Zeng, D., Zhang, X.-P., and Mei, T. (2022). The elements of end-to-end deep face recognition: A survey of recent advances. *ACM Comput. Surv.*, 54(10s).

Geng, C., Huang, S.-J., and Chen, S. (2021). Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Liang, X., Wang, X., Lei, Z., Liao, S., and Li, S. Z. (2017). Soft-margin softmax for deep classification. In *International Conference on Neural Information Processing*.

Liu, W., Wen, Y., Raj, B., Singh, R., and Weller, A. (2023). Sphereface revived: Unifying hyperspherical face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2458–2474.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). SphereFace: Deep hypersphere embedding for face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, W., Wen, Y., Yu, Z., and Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 507–516, New York, New York, USA. PMLR.

Mahdavi, A. and Carvalho, M. (2021). A survey on open set recognition. In *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 37–44.

Meng, Q., Zhao, S., Huang, Z., and Zhou, F. (2021). MagFace: A universal representation for face recognition and quality assessment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436.

Palechor, A., Bhoumik, A., and Günther, M. (2023). Large-scale open-set classification protocols for imagenet. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 42–51. CVF/IEEE.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. (2013). Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:623–656.

Wang, F., Cheng, J., Liu, W., and Liu, H. (2018a). Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930.

Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. (2017). Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1041–1049, New York, NY, USA. Association for Computing Machinery.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. (2018b). CosFace: Large margin cosine loss for deep face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 499–515, Cham. Springer International Publishing.

Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Learning face representation from scratch. *CoRR*, abs/1411.7923.

Zhang, X., Zhao, R., Qiao, Y., Wang, X., and Li, H. (2019). Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zheng, Y., Pal, D. K., and Savvides, M. (2018). Ring loss: Convex feature normalization for face recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5089–5097.